



# ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES Y DE TELECOMUNICACIÓN

Titulación:

INGENIERO EN INFORMÁTICA

Título del proyecto:

DESARROLLO DE UNA APLICACIÓN DE COMPRAVENTA  
DE ACCIONES: LABOLSADELALIGA

Javier Lapieza Fernández

Tutor: Eduardo Alfaro Larragueta

Pamplona, 7 de septiembre de 2011



<b>1. Introducción .....</b>	<b>6</b>
<b>1.1. Objetivos .....</b>	<b>6</b>
<b>1.2. Descripción del problema por parte del cliente.....</b>	<b>6</b>
<b>1.3. Requisitos .....</b>	<b>7</b>
1.3.1. Requisitos generales .....	7
1.3.2. Requisitos funcionales .....	8
1.3.3. Requisitos no funcionales.....	9
<b>1.4. Solución propuesta.....</b>	<b>9</b>
<b>1.5. Herramientas necesarias .....</b>	<b>10</b>
1.5.1. Servidor web Apache .....	10
1.5.2. Servidor de servlets Tomcat.....	10
1.5.3. AJAX.....	11
1.5.4. Motor de plantillas Apache Velocity .....	12
1.5.5. MySQL .....	13
1.5.6. Código HTML y XML.....	13
1.5.7. JavaScript.....	14
1.5.8. Java .....	15
1.5.9. Subversion.....	15
1.5.10. NetBeans .....	16
1.5.11. Otras herramientas.....	16
<b>2. Análisis .....</b>	<b>17</b>
<b>2.1. Casos de uso .....</b>	<b>17</b>
2.1.1. Visión general del sistema.....	17
2.1.2. Administración .....	18
2.1.3. Gestión del sitio.....	19
2.1.3.1. Textos .....	19
2.1.3.2. Noticias .....	20
2.1.4. Gestión de afiliados .....	21
2.1.4.1. Consultas afiliados .....	21
2.1.4.2. Aprobar afiliado .....	22
2.1.5. Blog.....	23
2.1.5.1. Nuevo Post .....	23
2.1.5.2. Editar Post.....	24
2.1.5.3. Moderar comentario.....	25
2.1.6. Gestor de Preguntas Frecuentes.....	26
2.1.6.1. Alta de pregunta frecuente.....	26
2.1.6.2. Baja de pregunta frecuente.....	27
2.1.6.3. Modificación de pregunta frecuente .....	27
2.1.6.4. Listado de preguntas frecuentes .....	28
2.1.7. Gestión de usuarios .....	29
2.1.7.1. Buscar usuarios.....	29
2.1.7.2. Datos de usuario .....	30
2.1.7.3. Modificar datos de usuario.....	30
2.1.8. Contabilidad .....	31
2.1.8.1. Consultas generales.....	31
2.1.8.2. Operaciones .....	32
2.1.8.3. Generar cuadernos de pago .....	32
2.1.9. Gestor de devoluciones.....	33
2.1.9.1. Alta de devolución .....	33
2.1.9.2. Baja de devolución .....	34
2.1.10. Visitante.....	35
2.1.10.1. Registro.....	35

2.1.10.2. Registro afiliados .....	36
2.1.11. Ayuda .....	37
2.1.11.1. Glosario .....	37
2.1.11.2. Preguntas frecuentes .....	38
2.1.11.3. Cómo juego .....	38
2.1.11.4. Buscador .....	39
2.1.11.5. Atención al cliente .....	39
2.1.12. Usuarios .....	40
2.1.12.1. Login .....	40
2.1.12.2. Recordar contraseña .....	41
2.1.12.3. Adquirir bolsas .....	42
2.1.12.4. Mercado Continuo .....	43
2.1.12.5. Cobrar mi dinero .....	43
2.1.13. Consultas competición .....	44
2.1.13.1. La clasificación .....	44
2.1.13.2. La jornada .....	45
2.1.13.3. Crea tu estrategia .....	46
2.1.13.4. Actualizar datos .....	46
2.1.14. Gestión de mi bolsa .....	48
2.1.14.1. Ver ofertas .....	48
2.1.14.2. Operar .....	49
2.1.14.3. Publicar ofertas .....	50
2.1.14.4. Mis movimientos .....	51
2.1.14.5. Eliminar ofertas .....	51
2.1.14.6. Recordar clave de firma .....	52
2.1.15. Gestión de alertas .....	53
2.1.15.1. Ver alertas .....	53
2.1.15.2. Crear alerta .....	54
2.1.15.3. Eliminar alerta .....	54
2.1.16. Mis datos .....	55
2.1.16.1. Mis datos .....	55
2.1.16.2. Cambiar datos personales .....	56
2.1.16.3. Cambiar datos de cuenta .....	56
2.1.17. Opinión .....	57
2.1.17.1. Comentar .....	57
2.1.17.2. Valorar .....	58
2.1.17.3. Consultar datos .....	58
<b>3. Diseño .....</b>	<b>59</b>
<b>3.1 Funcionamiento general .....</b>	<b>59</b>
<b>3.2 Modelo de datos .....</b>	<b>62</b>
3.2.1. Usuarios .....	62
3.2.2. Acciones .....	63
3.2.3. Alarmas .....	64
<b>4. Implementación .....</b>	<b>65</b>
<b>4.1. Interacción con la base de datos .....</b>	<b>65</b>
4.1.1. Plantillas para la generación de código .....	65
4.1.2. Código generado automáticamente .....	66
<b>4.2. Software propietario .....</b>	<b>66</b>
4.2.1. Servlets de peticiones de información .....	67
4.2.2. Utilidad de compraventa .....	67
4.2.3. Refresco de cotizaciones .....	68
4.2.4. Sistema de Alertas .....	68



4.2.5. Servlet de administración.....	68
<b>4.3. Proceso de desarrollo.....</b>	<b>69</b>
4.3.1. Proceso actual de desarrollo .....	69
4.3.2. Mejoras al proceso de desarrollo .....	70
<b>5. Implantación .....</b>	<b>71</b>
<b>5.1. Estructura del sistema .....</b>	<b>71</b>
5.1.1. Servidor Apache.....	72
5.1.2. Servidor Tomcat .....	73
5.1.3. Base de datos MySql.....	73
5.1.4. Escalabilidad .....	74
5.1.5 Alta disponibilidad.....	75
<b>5.2. Parámetros de configuración .....</b>	<b>77</b>
<b>6. Pruebas.....</b>	<b>79</b>
<b>6.1. Pruebas de carga .....</b>	<b>79</b>
<b>6.2. Análisis teórico del ancho de banda .....</b>	<b>79</b>
6.2.1. Estimación del número de usuarios concurrentes .....	79
6.2.2. Influencia del refresco en el ancho de banda.....	83
6.2.3. Conclusiones.....	84
<b>6.3 Auditorias.....</b>	<b>86</b>
<b>7. Conclusiones y líneas futuras .....</b>	<b>87</b>
<b>7.1. Conclusiones.....</b>	<b>87</b>
<b>7.2. Repercusión en medios.....</b>	<b>88</b>
<b>7.3. Líneas futuras.....</b>	<b>88</b>
<b>8. Bibliografía .....</b>	<b>90</b>
<b>ANEXO I. MANUAL DE USUARIO .....</b>	<b>91</b>
<b>1. Pantalla principal .....</b>	<b>91</b>
<b>2. Registro .....</b>	<b>92</b>
2.1. Formulario de registro .....	92
2.2. Adquisición de bolsas.....	93
<b>3. Acceso .....</b>	<b>95</b>
3.1. Formulario de acceso .....	95
3.2. Recuperar contraseña .....	95
<b>4. Mi bolsa .....</b>	<b>96</b>
4.1 El menú.....	96
4.2 Comprar y vender acciones .....	97
4.3. Pantallas de equipo .....	98
4.4. Recuperar clave de firma.....	99
<b>5. Mis movimientos .....</b>	<b>100</b>
<b>6. Mis datos.....</b>	<b>102</b>
<b>7. Mis Alertas .....</b>	<b>104</b>
<b>8. Crea tu estrategia .....</b>	<b>105</b>
<b>9. Cobrar mi dinero .....</b>	<b>106</b>
<b>10. Mercado Continuo.....</b>	<b>107</b>
<b>11. Ayuda .....</b>	<b>108</b>
11.1 Preguntas frecuentes .....	108
11.2 Glosario.....	109
11.3 Atención al cliente.....	110

## **1. Introducción**

El mundo de Internet evoluciona muy rápido, cada vez más hogares tienen conexiones con mayor ancho de banda. Actualmente se realiza un número elevado de tareas cotidianas a través de Internet y, ésta, parece ser una tendencia al alza. Por todo esto no parece raro que también el ocio tenga un hueco en la red.

En los últimos el mundo de los juegos on-line, especialmente las apuestas, ha experimentado una gran progresión donde la oferta es amplísima. Juegos de estrategia para construir tu ciudad, juegos sociales, juegos de cartas adaptados...

No obstante, en lo relacionado con los deportes, pese a existir mucha oferta la variedad de la misma es limitada, básicamente se repite lo mismo una y otra vez: apuestas deportivas, alineaciones, porras...

Por otro lado, la banca on-line se está convirtiendo en algo cotidiano y junto con ella la compraventa de acciones en la red también se ha convertido en algo habitual con las últimas tecnologías.

Este proyecto pretende unir de alguna manera estos dos conceptos: las apuestas deportivas y el mercado de valores, o la bolsa, para crear un juego totalmente novedoso.

### **1.1. Objetivos**

El objeto de este proyecto es la implementación de una aplicación de compraventa de acciones de los equipos de fútbol de la liga de fútbol profesional cuyo núcleo pueda exportarse a otras ligas o incluso deportes.

### **1.2. Descripción del problema por parte del cliente**

El cliente plantea la idea de la bolsa de la liga como concepto aún sin desarrollar. El cliente no está familiarizado con el mundo del software y tampoco posee conocimientos sobre internet más allá del nivel de usuario.

Tras la propuesta inicial se realizan varias reuniones en las que se definen unos requisitos generales de la aplicación, sin centrarse aún en funcionalidades concretas de la misma. Estas reuniones sirven para definir el alcance del proyecto y las siguientes características generales.

El cliente solicita que la aplicación sea accesible vía web, por la universalidad del medio, por lo tanto la solución tiene que adaptarse a este requisito que marca de manera significativa la fase de diseño del sistema.

Se espera que el producto final sea robusto, escalable y eficiente ya que se espera dar servicio a un gran número de usuarios de manera simultánea.

El sistema tiene que procesar pagos y a trabajar con dinero real, por lo que todas las operaciones con el mismo deben ser robustas e ir lo suficientemente aseguradas frente a errores.

Otra imposición resultante de trabajar con dinero es la seguridad de los datos referentes a los usuarios que acceden al sistema. El acceso al mismo y cualquier operación que implique alteraciones sobre el capital de un usuario deben ser seguros. Los datos de acceso y de operación deben estar suficientemente protegidos. Es condición necesaria que el sistema supere una auditoria de seguridad por parte de una empresa externa antes de su publicación.

El almacenamiento de los datos sobre los usuarios deberá ajustarse a la LOPD.

Adaptabilidad: una vez se tenga un sistema estable para la liga española debería ser sencillo adaptarlo para trabajar en otras ligas e incluso en otros deportes, por lo tanto el sistema debe ser adaptable y escalable.

Compatibilidad: uno de los objetivos es llegar al máximo número de usuarios posible, por lo que el sistema debe ser compatible con el mayor número posible de plataformas y sistemas operativos.

Frecuencia de actualización: el cliente insiste en que la frecuencia de actualización de todos los datos de la web debe ser lo más alta posible. En sus palabras debe actualizarse “en tiempo real”.

### **1.3. Requisitos**

Tras la descripción general del problema se mantienen reuniones con el cliente y otros agentes involucrados para definir el sistema más exhaustivamente. De estas reuniones se extraen los siguientes requisitos.

#### **1.3.1. Requisitos generales**

Para poder participar en el juego será necesario realizar un pago, este pago se realizará empleando una pasarela de pago virtual (TPV virtual) sin salir de la aplicación. Una vez realizado el pago, el usuario dispone de un paquete de acciones y de una cantidad determinada de efectivo para operar en el mercado de la manera que estime más conveniente.

La compraventa de acciones se realiza entre usuarios y tiene que cumplir unas normas de mercado, si a un usuario no le satisfacen las ofertas existentes podrá publicar una oferta que le resulte más beneficiosa y esperar a que otro usuario decida aceptarla.

Todas las operaciones de mercado que realice el usuario deben ser validadas mediante una clave de firma. En todos los lugares donde sea necesario introducir esta clave el usuario tendrá la posibilidad de pedir que le sea recordada.

El usuario podrá comprar y vender acciones desde todas las pantallas relacionadas con el juego o la competición.

El cliente tiene las diferentes secciones y la estructura de la página muy claras.

“Mi Bolsa” donde los usuarios podrán consultar el estado actual de su bolsa. Esta será la página principal de la zona de usuarios. Es el centro de operaciones donde tiene lugar el mayor número de transacciones.

“Mis movimientos” donde los usuarios podrán consultar todos los movimientos realizados y pendientes de realizar. Los usuarios pueden consultar compras y ventas, realizadas y por realizar, por equipos, por fechas...

“Mis datos” donde los usuarios pueden consultar y modificar los datos de su cuenta. Además el usuario podrá consultar los datos referentes a la competición de manera rápida la clasificación, el calendario, los resultados...

El usuario y el visitante tendrán a su disposición un sistema de ayuda lo más completo posible y además se les facilitará un formulario para enviar un correo electrónico en caso de que no pueda resolver su duda.

La aplicación ofrecerá servicios adicionales a los usuarios tales como configuración de alertas de correo, información de clasificaciones y resultados, noticias referentes a la competición...

Los datos referentes a la competición así como las cotizaciones deben refrescarse con la mayor frecuencia posible. El cliente solicita “tiempo real”.

Además la aplicación tendrá un sistema de administración o backoffice de forma que el administrador o los administradores dispongan de una herramienta de control tanto global (estado de la aplicación) como individual (datos y estado de la bolsa de un usuario). También incluirá un editor de contenidos de la web, textos, entradas en blogs, moderación de comentarios, gestión de promociones... Además de una herramienta para llevar un control de la contabilidad y la facturación.

El acceso al backoffice podrá ser total o parcial (previa alta por parte de los administradores) permitiendo a terceros acceso a ciertos datos que puedan ser interesantes para ellos.

### **1.3.2. Requisitos funcionales**

La información ofrecida al usuario debe mantenerse actualizada continuamente, el periodo de refresco debe ser configurable.

El sistema debe soportar múltiples idiomas de manera sencilla. El sistema debe ser compatible con todos los navegadores actuales.

El sistema debe estar preparado para ser fácilmente adaptable a otras ligas o competiciones.

El sistema debe asegurar y proteger la coherencia de los datos del mismo, especialmente en todo lo que se refiere a operaciones de compraventa de acciones y a los pagos realizados por parte de los clientes.

El sistema almacenará toda la información sensible encriptada y solo almacenará información sensible cuando sea necesario.

### 1.3.3. Requisitos no funcionales

La aplicación deberá situarse lo más alto posible en las páginas de resultados de los buscadores más populares.

Las direcciones de los documentos de la página (URLs) de la parte pública de la misma serán limpias y fácilmente interpretables por humanos para facilitar su memorización y favorecer el posicionamiento en buscadores.

El diseño de la página debe ser sencillo e intuitivo.

El acceso a los documentos, plantillas y código, debe estar suficientemente protegido en todas las fases del desarrollo y en los servidores de producción.

Se trabajará sobre un sistema de control de versiones.

El entorno en el que funcionará el sistema será un entorno similar a LAMP (Linux, Apache, MySQL y PHP)

### 1.4. Solución propuesta

Se propone realizar la implementación del juego en una web dinámica que contenga tanto el juego como todo el sistema de administración.

Para dar respuesta a esta necesidad dinámica de la capa del sitio y alojar el juego en su conjunto, se propone diseñar el sitio siguiendo el paradigma “MVC” (“Model-View-Controller” o “Modelo-Vista-Controlador”) usando componentes de programación JAVA/J2EE.

En este marco de programación, los programas se estructurarán en tres capas diferenciadas. Por un lado está la capa de presentación o vistas, por otro el modelo de datos que suele estar soportado en un gestor de base de datos, y una tercera y última capa que es la lógica de negocio y control de la usabilidad del programa según vaya interactuando con los usuarios.

El modelo se ha diseñado en un esquema entidad-relación implementado en un gestor de base de datos relacional, que garantiza la persistencia de los datos generados. Concretamente se usará MySQL como gestor de base de datos relacional como se indica en los requisitos, al ser un sistema gestor libre.

Finalmente la capa de control se implementará en un conjunto de servlets Java, que son los encargados de obtener los datos del modelo, generar las vistas, y controlar la lógica del juego. Se puede diferenciar cuatro tipos de servlets de acuerdo a la funcionalidad que proporcionan:

- Pantallas públicas (páginas del sitio abiertas a navegantes anónimos)
- Pantallas del juego (para usuarios registrados, previa identificación)
- Datos en formato XML para su consumo por parte de scripts (AJAX)
- Pantallas del administrador del sistema

## 1.5. Herramientas necesarias

### 1.5.1. Servidor web Apache

El servidor HTTP Apache es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux...), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que Behelendorf quería que tuviese la connotación de algo que es firme y enérgico pero no agresivo y la tribu Apache fue la última en rendirse al que, pronto, se convertiría en gobierno de EEUU y, en esos momentos, la preocupación de su grupo era que llegasen las empresas y "civilizasen" el paisaje que habían creado los primeros ingenieros de internet. Además Apache consistía solamente en un conjunto de parches a aplicar al servidor de NCSA. Era, en inglés, *a patchy server* (un servidor "parcheado").

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation.

Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años.

La mayoría de las vulnerabilidades de la seguridad descubiertas y resueltas tan sólo pueden ser aprovechadas por usuarios locales y no remotamente. Sin embargo, algunas se pueden accionar remotamente en ciertas situaciones o explotar por los usuarios locales malévolos en las disposiciones de recibimiento compartidas que utilizan PHP como módulo de Apache. En nuestro sistema no se utilizará PHP.

### 1.5.2. Servidor de servlets Tomcat

Tomcat es un contenedor de Servlets con un entorno JSP. Un contenedor de Servlets es un shell de ejecución que maneja e invoca servlets por cuenta del usuario.

Se pueden dividir los contenedores de Servlets en:

1. Contenedores de Servlets Stand-alone (Independientes): Estos son una parte integral del servidor web. Este es el caso cuando usando un servidor web basado en Java, por ejemplo, el contenedor de servlets es parte de JavaWebServer (actualmente sustituido por iPlanet). Este el modo por defecto usado por Tomcat. Sin embargo, la mayoría de los servidores, no están basados en Java, por lo que surgen los dos siguientes tipos de contenedores
2. Contenedores de Servlets dentro-de-Proceso: El contenedor Servlet es una combinación de un plugin para el servidor web y una implementación de contenedor Java. El plugin del servidor web abre una JVM (Máquina Virtual

Java) dentro del espacio de direcciones del servidor web y permite que el contenedor Java se ejecute en él. Si una cierta petición debería ejecutar un servlet, el plugin toma el control sobre la petición y lo pasa al contenedor Java (usando JNI). Un contenedor de este tipo es adecuado para servidores multi-thread de un sólo proceso y proporciona un buen rendimiento pero está limitado en escalabilidad

3. Contenedores de Servlets fuera-de-proceso: El contenedor Servlet es una combinación de un plugin para el servidor web y una implementación de contenedor Java que se ejecuta en una JVM fuera del servidor web. El plugin del servidor web y el JVM del contenedor Java se comunican usando algún mecanismo IPC (normalmente sockets TCP/IP). Si una cierta petición debería ejecutar un servlet, el plugin toma el control sobre la petición y lo pasa al contenedor Java (usando IPCs). El tiempo de respuesta en este tipo de contenedores no es tan bueno como el anterior, pero obtiene mejores rendimientos en otras cosas (escalabilidad, estabilidad...).

Tomcat puede utilizarse como un contenedor solitario (principalmente para desarrollo y depuración) o como plugin para un servidor web existente (actualmente se separan los servidores Apache, IIS y Netscape). Esto significa que siempre que se despliegue Tomcat hay que decidir cómo usarlo y, si se escogen las opciones 2 o 3, también será necesario instalar un adaptador de servidor web.

En este proyecto se usará el segundo tipo de servlet integrado con el servidor web Apache que ya se ha presentado.

### 1.5.3. AJAX

Ajax, acrónimo de *Asynchronous JavaScript And XML* (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (*Rich Internet Applications*). Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

Ajax es una tecnología asíncrona, en el sentido de que los datos adicionales se requieren al servidor y se cargan en segundo plano sin interferir con la visualización ni el comportamiento de la página. JavaScript es el lenguaje interpretado (scripting language) en el que normalmente se efectúan las funciones de llamada de Ajax mientras que el acceso a los datos se realiza mediante *XMLHttpRequest*, objeto disponible en los navegadores actuales. En cualquier caso, no es necesario que el contenido asíncrono esté formateado en XML.

Ajax es una técnica, válida para múltiples plataformas y utilizable en muchos sistemas operativos y navegadores, dado que está basado en estándares abiertos como JavaScript y Document Object Model (DOM).

El sistema aprovecha las características de AJAX para mejorar el rendimiento de la parte dinámica de la página además de resultar crucial en la actualización de datos.



#### 1.5.4. Motor de plantillas Apache Velocity

Velocity es un motor de plantillas basado en Java. Permite a los diseñadores de páginas hacer referencia a métodos definidos dentro del código Java. Los diseñadores Web pueden trabajar en paralelo con los programadores Java para desarrollar sitios de acuerdo al modelo de Modelo-Vista-Controlador (MVC), permitiendo que los diseñadores se concentren únicamente en crear un sitio bien diseñado y que los programadores se encarguen solamente de escribir código de primera calidad. Velocity separa el código Java de las páginas Web, haciendo el sitio más mantenible a largo plazo y presentando una alternativa viable a Java Server Pages (JSP) o PHP.

Velocity se puede utilizar para crear páginas web, SQL, PostScript y cualquier otro tipo de salida de plantillas. Se puede utilizar como una aplicación independiente para generar código fuente y reportes o como un componente integrado en otros sistemas. Cuando Velocity esté completo proveerá servicios para el conjunto de aplicaciones web de Turbine. Velocity+Turbine proveerá un servicio de plantillas que facilitará el desarrollo de aplicaciones web de acuerdo a un verdadero modelo MVC.

El Lenguaje de Plantillas de Velocity (VTL) fue creado para proveer la manera más fácil, simple y limpia de incorporar contenido dinámico dentro de una página web. Incluso un desarrollador de páginas web con poca y ninguna experiencia puede rápidamente ser capaz de utilizar VTL para incluir contenido dinámico en un sitio web.

VTL usa *referencias* para incluir contenido dinámico dentro de un sitio web. Una variable es un tipo de referencia que puede referirse a algo definido dentro del código Java u obtener su valor de un *enunciado* VTL en la página misma. A continuación hay un ejemplo de un enunciado VTL que se puede incluir en un documento HTML:

```
#set( $a = "Velocity" )
```

Una vez un valor ha sido asignado a una variable, se puede hacer referencia a la variable en cualquier lugar dentro del documento HTML. En el ejemplo siguiente, un valor es asignado a la variable *\$foo*, que luego es referenciada.

```
<html>
  <body> #set( $foo = "Velocity" )
        ¡Hola Mundo $foo!
  </body>
</html>
```

El resultado es una página web que imprime "¡Hola Mundo Velocity!"

Es posible unir estas plantillas con el código del servlet a partir del motor velocity. Cada petición crea un contexto donde colocar estas variables. Desde este contexto se extraen las variables que se colocan en las plantillas.

Se puede colocar prácticamente cualquier tipo de objeto en las plantillas desde el servlet y acceder a sus atributos.

```
context.put("variable_usuario", usuario);
```



Si se coloca el objeto usuario en la variable `variable_usuario`, por ejemplo, sería posible acceder a cualquier atributo del mismo de manera sencilla.

```
<html>
  <body>
    <p> Nombre: $variable_usuario.nombre </p> <br />
    <p> Apellido: $variable_usuario.apellido </p> <br />
  </body>
</html>
```

El resultado es una página con el nombre y apellido del usuario.

En este proyecto se emplea Velocity en la generación de plantillas HTML, JavaScript y CSS.

### 1.5.5. MySQL

MySQL es un sistema de gestión de bases de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB —desde enero de 2008 una subsidiaria de Sun Microsystems y ésta a su vez de Oracle Corporation desde abril de 2009— desarrolla MySQL como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Al contrario de proyectos como Apache, donde el software es desarrollado por una comunidad pública y los derechos de autor del código están en poder del autor individual, MySQL es patrocinado por una empresa privada, que posee el copyright de la mayor parte del código.

Esto es lo que posibilita el esquema de licenciamiento anteriormente mencionado. Además de la venta de licencias privativas, la compañía ofrece soporte y servicios. Para sus operaciones contratan trabajadores alrededor del mundo que colaboran vía Internet.

### 1.5.6. Código HTML y XML

HTML es el lenguaje de marcado predominante para la elaboración de páginas web. Es usado para describir la estructura y el contenido en forma de texto, así como para complementar el texto con objetos tales como imágenes. HTML se escribe en forma de etiquetas rodeadas por corchetes angulares (<,>). HTML también puede describir, hasta un cierto punto, la apariencia de un documento, y puede incluir un *script* (por ejemplo JavaScript), el cual puede afectar el comportamiento de navegadores web y otros procesadores de HTML.

HTML también es usado para referirse al contenido del tipo de MIME `text/html` o todavía más ampliamente como un término genérico para el HTML, ya sea en forma descendida del XML (como XHTML 1.0 y posteriores) o en forma descendida directamente de SGML (como HTML 4.01 y anteriores).

XML, siglas en inglés de *eXtensible Markup Language* ('lenguaje de marcas extensible'), es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Algunos de estos lenguajes que usan XML para su definición son XHTML, SVG, MathML.

XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

En nuestro sistema XML desempeña un papel muy importante ya que el tráfico de información se basa en gran medida en el uso de XML.

### 1.5.7. JavaScript

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico.

Se utiliza principalmente en su forma del lado del cliente (client-side), implementado como parte de un navegador web permitiendo mejoras en la interfaz de usuario y páginas web dinámicas, aunque existe una forma de JavaScript del lado del servidor (Server-side JavaScript o SSJS). Su uso en aplicaciones externas a la web, por ejemplo en documentos PDF, aplicaciones de escritorio (mayoritariamente widgets) es también significativo.

JavaScript se diseñó con una sintaxis similar al C, aunque adopta nombres y convenciones del lenguaje de programación Java. Sin embargo Java y JavaScript no están relacionados y tienen semánticas y propósitos diferentes.

Todos los navegadores modernos interpretan el código JavaScript integrado en las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del Document Object Model (DOM).

Tradicionalmente se venía utilizando en páginas web HTML para realizar operaciones y únicamente en el marco de la aplicación cliente, sin acceso a funciones del servidor. JavaScript se interpreta en el agente de usuario, al mismo tiempo que las sentencias van descargándose junto con el código HTML.

### 1.5.8. Java

Java es un lenguaje de programación orientado a objetos, desarrollado por Sun Microsystems a principios de los años 90. El lenguaje en sí mismo toma mucha de su sintaxis de C y C++, pero tiene un modelo de objetos más simple y elimina herramientas de bajo nivel, que suelen inducir a muchos errores, como la manipulación directa de punteros o memoria.

Las aplicaciones Java están típicamente compiladas en un *bytecode*, aunque la compilación en código máquina nativo también es posible. En el tiempo de ejecución, el *bytecode* es normalmente interpretado o compilado a código nativo para la ejecución, aunque la ejecución directa por hardware del *bytecode* por un procesador Java también es posible.

La implementación original y de referencia del compilador, la máquina virtual y las bibliotecas de clases de Java fueron desarrolladas por Sun Microsystems en 1995. Desde entonces, Sun ha controlado las especificaciones, el desarrollo y evolución del lenguaje a través del Java Community Process, si bien otros han desarrollado también implementaciones alternativas de estas tecnologías de Sun, algunas incluso bajo licencias de software libre.

Entre diciembre de 2006 y mayo de 2007, Sun Microsystems liberó la mayor parte de sus tecnologías Java bajo la licencia GNU GPL, de acuerdo con las especificaciones del Java Community Process, de tal forma que prácticamente todo el Java de Sun es ahora software libre (aunque la biblioteca de clases de Sun que se requiere para ejecutar los programas Java aún no lo es).

### 1.5.9. Subversion

Subversion es un sistema de control de versiones diseñado específicamente para reemplazar al popular CVS. Es software libre bajo una licencia de tipo Apache/BSD y se le conoce también como *svn* por ser el nombre de la herramienta utilizada en la línea de órdenes.

Una característica importante de Subversion es que, a diferencia de CVS, los archivos versionados no tienen cada uno un número de revisión independiente, en cambio, todo el repositorio tiene un único número de versión que identifica un estado común de todos los archivos del repositorio en un instante determinado.

Subversion puede acceder al repositorio a través de redes, lo que le permite ser usado por personas que se encuentran en distintas computadoras. A cierto nivel, la posibilidad de que varias personas puedan modificar y administrar el mismo conjunto de datos desde sus respectivas ubicaciones fomenta la colaboración. Se puede progresar más rápidamente sin un único conducto por el cual deban pasar todas las modificaciones. Y puesto que el trabajo se encuentra bajo el control de versiones, no hay razón para temer por que la calidad del mismo vaya a verse afectada.

### **1.5.10. NetBeans**

NetBeans es un entorno de desarrollo, hecho principalmente para el lenguaje de programación Java. Existe además un número importante de módulos para extender el NetBeans IDE. NetBeans IDE es un producto libre y gratuito sin restricciones de uso.

NetBeans es un proyecto de código abierto de gran éxito con una gran base de usuarios, una comunidad en constante crecimiento, y con cerca de 100 socios en todo el mundo. Sun Microsystems fundó el proyecto de código abierto NetBeans en junio de 2000 y continúa siendo el patrocinador principal de los proyectos.

La plataforma NetBeans permite que las aplicaciones sean desarrolladas a partir de un conjunto de componentes de software llamados *módulos*. Un módulo es un archivo Java que contiene clases de java escritas para interactuar con las APIs de NetBeans y un archivo especial (manifest file) que lo identifica como módulo. Las aplicaciones construidas a partir de módulos pueden ser extendidas agregándole nuevos módulos. Debido a que los módulos pueden ser desarrollados independientemente, las aplicaciones basadas en la plataforma NetBeans pueden ser extendidas fácilmente por otros desarrolladores de software.

### **1.5.11. Otras herramientas**

Aparte de todos los lenguajes y herramientas mencionados también se han empleado otras tecnologías como Flash o plataformas como Adobe Photoshop y Macromedia Dreamweaver para diseño e implementación de páginas web. Estas herramientas son tratadas en un grupo aparte ya que son utilizadas mayoritariamente por otros miembros del equipo de trabajo.

## 2. Análisis

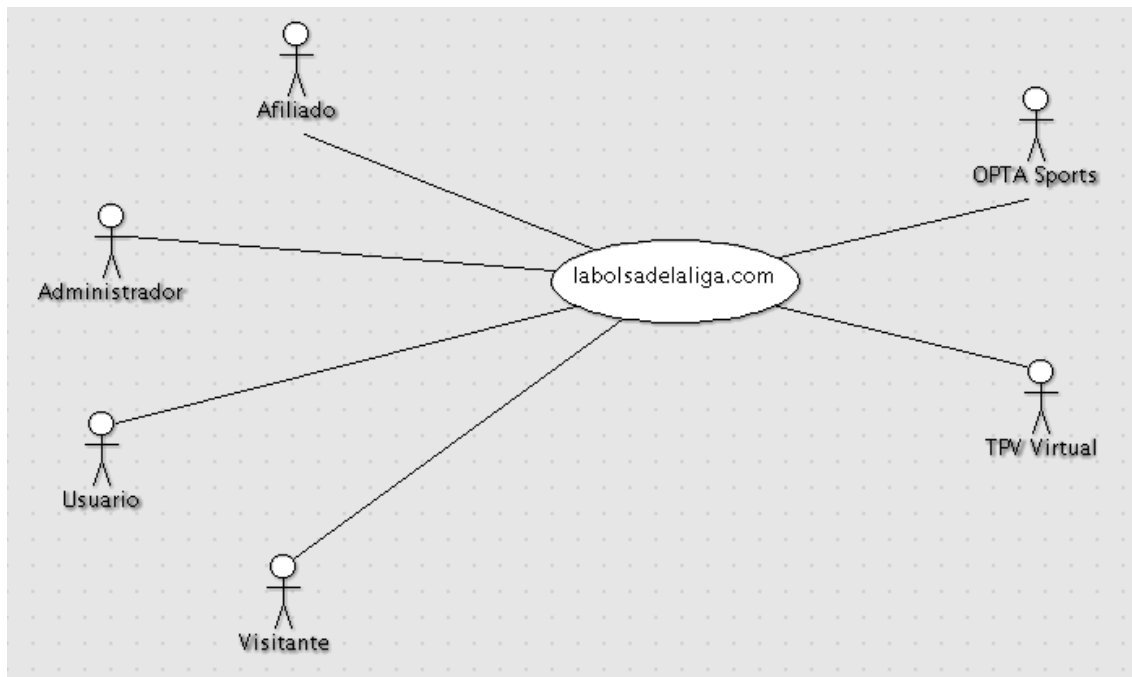
En esta sección se desglosan los casos de uso que definen la funcionalidad del sistema. Estos casos de uso son el resultado de diversas reuniones.

Tras las primeras reuniones que sirvieron para definir el alcance del proyecto, se realizaron reuniones involucrando a más personal, para definir poco a poco las diferentes secciones de la aplicación y sus funcionalidades concretas. En el transcurso de estas reuniones se emplearon prototipos sin funcionalidad completa donde se fueron refinando los procesos.

### 2.1. Casos de uso

En esta sección se muestran los casos de uso del sistema y sus sucesivas iteraciones. Cuando se alcancen casos de uso refinados por completo se definirán precondiciones, postcondiciones y flujos básico y alternativo.

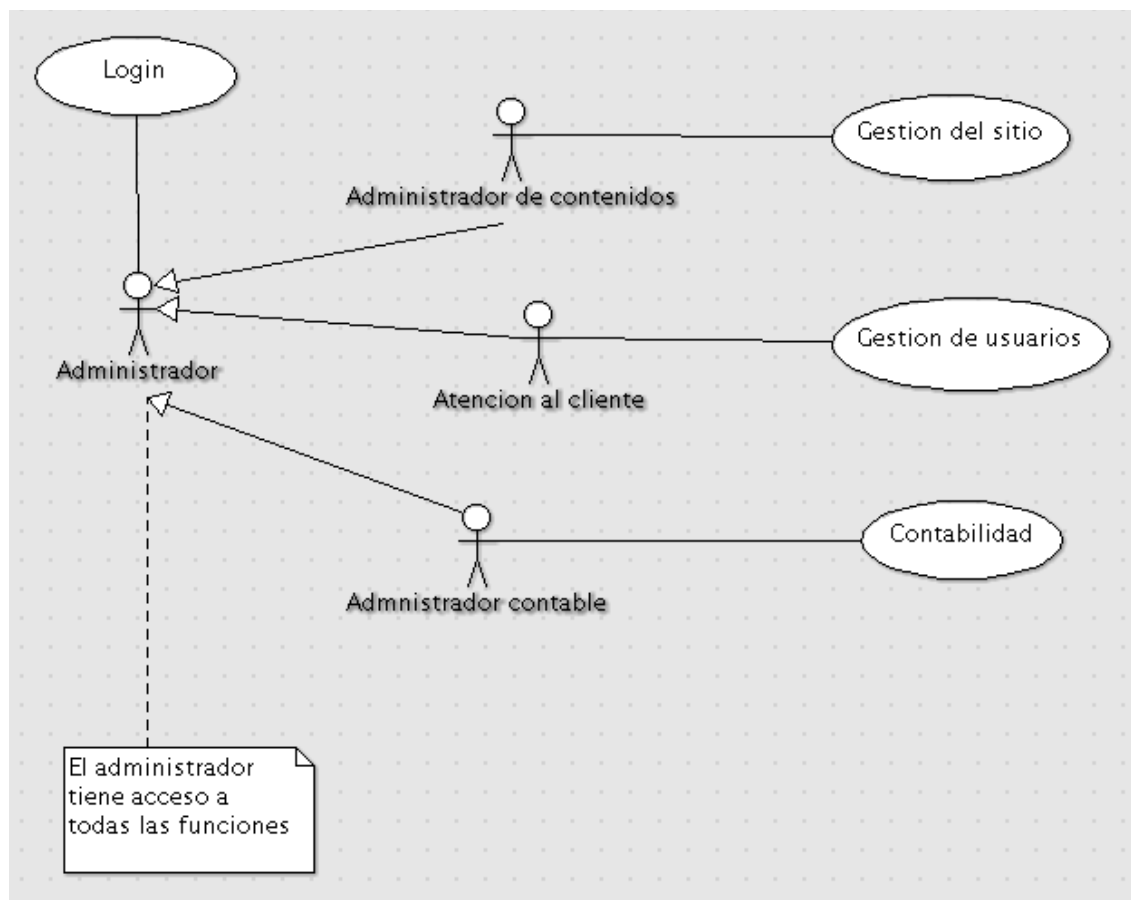
#### 2.1.1. Visión general del sistema



El sistema puede ser utilizado por diferentes tipos de usuario. Por un lado están los visitantes de la página web, estos si introducen usuario y contraseña se convierten en usuarios, o en afiliados, en función de que formulario de ingreso utilicen. Por otro lado están los administradores.

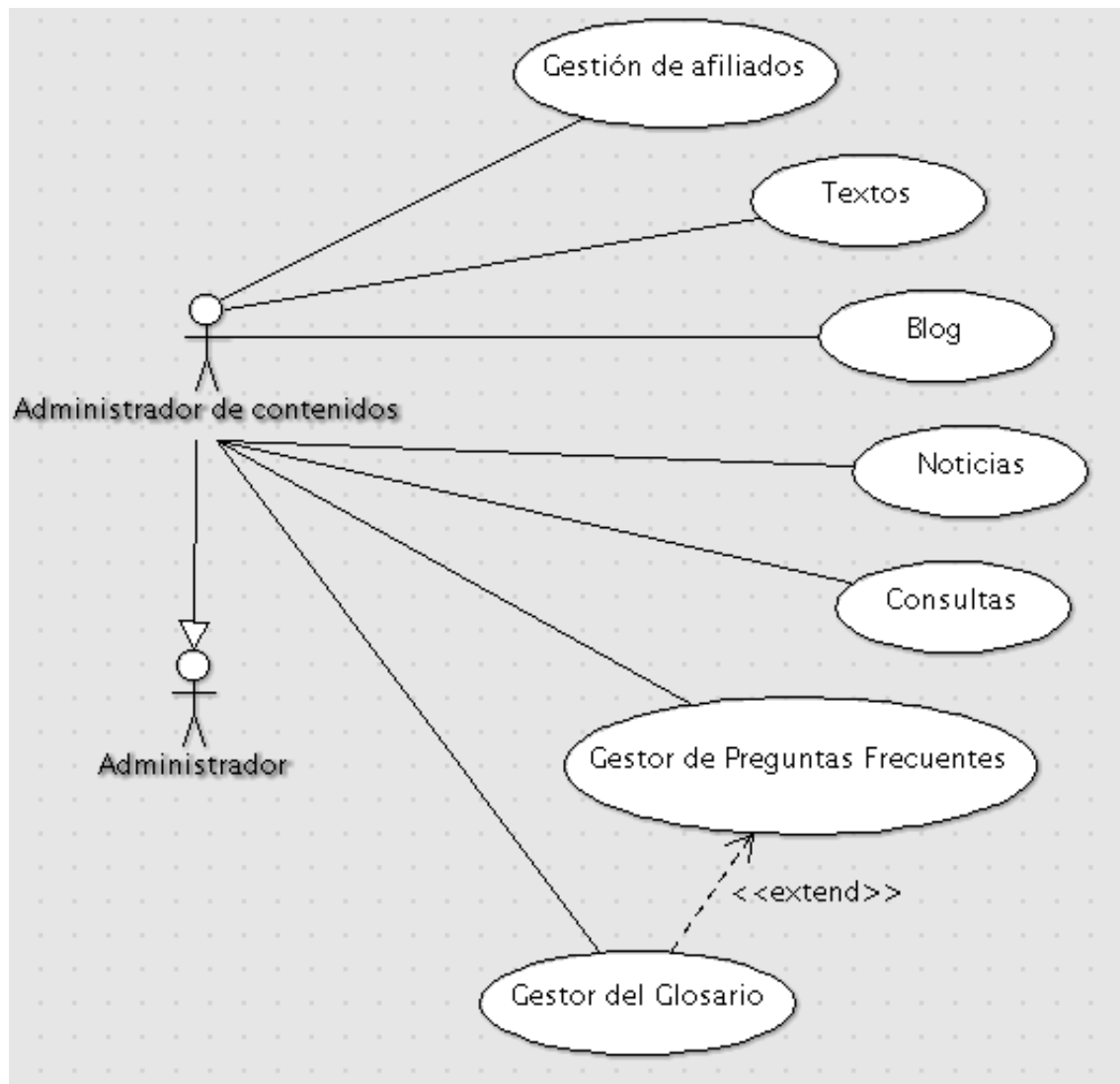
Por último, se puede ver que existen otros dos actores. En este caso se trata de dos entidades externas: OPTA Sports, que es una empresa que nos facilita los datos relativos a la competición de manera continua y asincrónica y el TPV Virtual, que se encarga de procesar los pagos de los clientes.

### 2.1.2. Administración



La sección de administración se divide en tres secciones principales, gestión del sitio, gestión de usuarios y contabilidad. Hay diferentes tipos de administrador, cada uno con acceso a unas zonas determinadas, los administradores generales tienen acceso a todas las zonas del administrador. Todos los administradores acceden a través de un login común que les redirige a su página de inicio correspondiente.

### 2.1.3. Gestión del sitio



La gestión del sitio es la sección más amplia del administrador. Desde aquí un administrador sin conocimientos informáticos debe ser capaz de añadir y modificar contenido en la web. La sección de ayuda permite gestionar las preguntas frecuentes y el glosario, teniendo en cuenta los requisitos multi-idioma. La sección de textos permite modificar cualquier texto de la web en cualquiera de los idiomas que estén dados de alta. El blog permite la gestión de entradas en el mismo, moderación de comentarios... La sección de noticias permite importar noticias manualmente desde las fuentes previamente seleccionadas y finalmente la sección de consultas permite realizar consultas generales sobre el estado del sistema.

#### 2.1.3.1. Textos

##### Descripción:

Un administrador puede modificar y crear nuevos textos para la web. La inclusión de estos textos en las páginas, no obstante, se realizará por la vía habitual editando el código HTML.

**Precondición:**

El usuario es administrador y tiene permisos suficientes.

**Postcondición:**

Se han realizado los cambios en los ficheros de texto correspondientes.

No existen ficheros con el mismo nombre.

No existen referencias con el mismo nombre dentro del mismo fichero.

**Flujo básico:**

1. El administrador escoge un fichero de textos que quiere modificar a través de un desplegable. Puede elegir crear un nuevo fichero.
2. El usuario elige entre las referencias existentes la que quiere modificar, o crea una nueva.
3. El usuario introduce un valor para esa referencia.(puede estar vacía)
4. Se guarda el fichero de texto.

**Flujo alternativo:**

- 1.1. Error de conexión con sistema de ficheros.
- 1.2. Mensaje de error.
- 1.3. No se carga el desplegable.

- 1.1 El fichero no tiene permisos suficientes
- 1.2 Mensaje de error.
- 1.3 No se cargan las referencias.

- 4.1 No se tienen permisos de escritura.
- 4.2 Mensaje de error.

2.1.3.2. Noticias

**Descripción:**

Permite realizar cargas de noticias fuera del horario programado. El sistema se conecta con los feeds RSS de noticias predefinidos y vuelca su contenido en la base de datos.

**Precondición:**

El usuario es administrador y tiene permisos suficientes.

**Postcondición:**

Se cargan noticias nuevas (si las hay) y sus imágenes.

No hay noticias repetidas

**Flujo básico:**

1. El administrador hace clic en cargar noticias.
2. Se establece conexión con el feed RSS.
3. Se guardan las noticias nuevas (con un título distinto a las existentes).
4. Se descargan las imágenes al servidor.

**Flujo alternativo:**

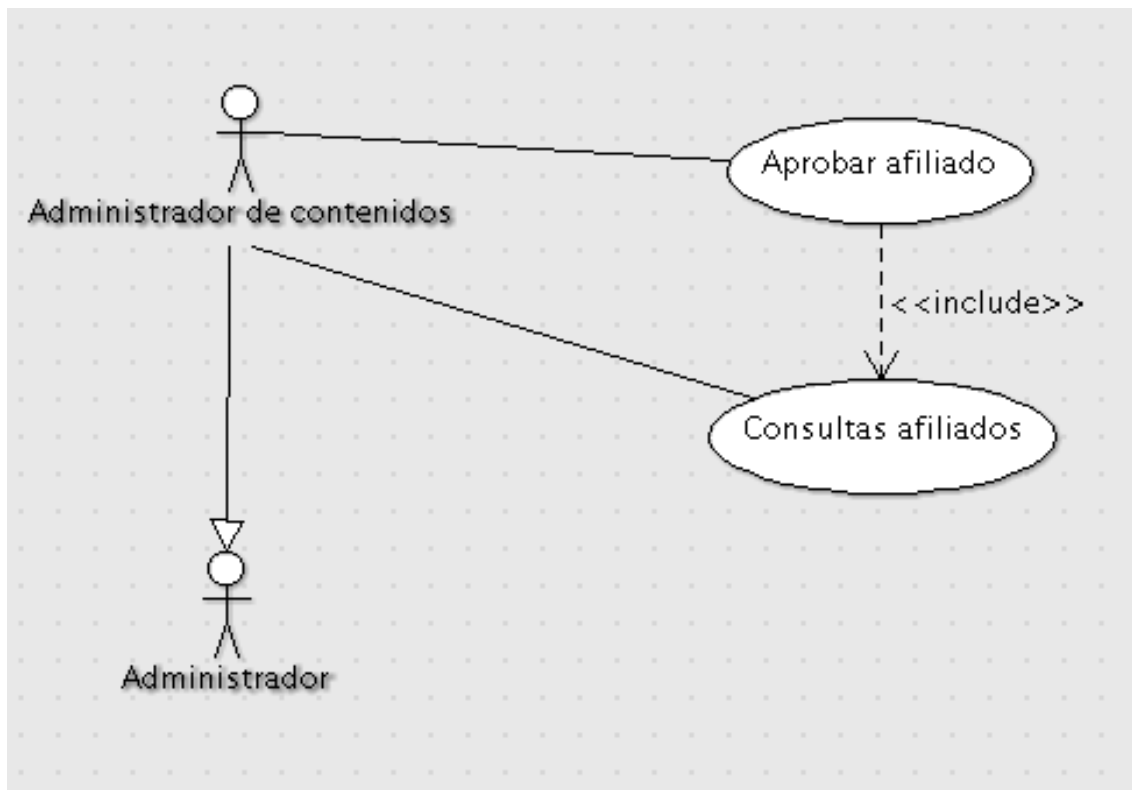
- 2.1 La dirección indicada no existe
- 2.2 Mensaje de error.
- 2.3 Se detiene la ejecución.



- 3.1 Error de conexión con la base de datos.
- 3.2 Mensaje de error.
- 3.3 Se detiene la ejecución.

- 4.1 La URL de la imagen es incorrecta.
- 4.2 No se descarga esa imagen.

#### 2.1.4. Gestión de afiliados



El administrador de contenidos puede consultar las peticiones de afiliación y aprobarlas desde esta página, además puede realizar consultas sobre los afiliados ya aprobados (visitas desde su web, bolsas vendidas...).

##### 2.1.4.1. Consultas afiliados

###### Descripción:

La pantalla principal muestra un listado con los afiliados que incluye: visitas, bolsas vendidas, página de afiliación y nombre de usuario.

###### Precondición:

El usuario es administrador y tiene permisos suficientes.

###### Postcondición:

Se muestran los datos pertinentes.

###### Flujo básico:

1. El administrador accede a la sección Afiliados.
2. Se obtienen los datos necesarios desde la base de datos.

**Flujo alternativo:**

- 2.1 Error de conexión con la base de datos.
- 2.2 Mensaje de error.
- 2.3 Se muestra la página vacía.

2.1.4.2. Aprobar afiliado

**Descripción:**

Un administrador da de alta a un afiliado asignándole un nombre de usuario, una contraseña y un id de tracking.

**Precondición:**

El usuario es administrador y tiene permisos suficientes.  
Existe una petición de afiliación y el afiliado no está dado de alta

**Postcondición:**

El afiliado esta dado de alta.  
El nombre de usuario y su id de tracking son únicos.

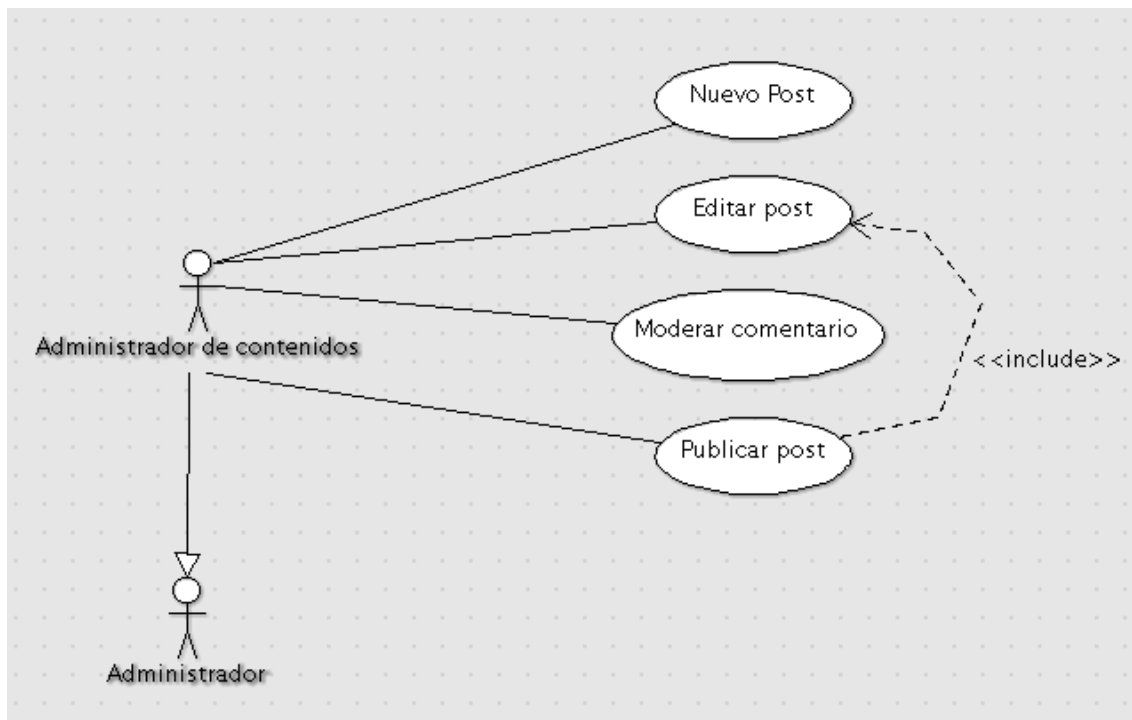
**Flujo básico:**

1. El administrador selecciona del listado de afiliados el afiliado que quiere dar de alta.
2. El administrador introduce los datos necesarios para dar de alta al afiliado: nombre de usuario, id de tracking, contraseña.
3. El sistema valida estos datos.
4. El afiliado es dado de alta.

**Flujo alternativo:**

- 1.1 Error de conexión con la base de datos.
- 1.2 Mensaje de error.
- 1.3 Los datos no se cargan.
  
- 3.1 Algún dato no es correcto.
- 3.2 Mensaje de error. Se solicita que se vuelvan a introducir los datos.
- 3.3 Se reintenta.
  
- 4.1 Error de conexión con la base de datos
- 4.2 Mensaje de error.
- 4.3 Se detiene la ejecución.

## 2.1.5. Blog



La sección del blog permite la gestión completa de posts (pueden dejarse a medio editar, programar publicaciones...) además de la consulta y moderación de comentarios sobre los distintos posts.

### 2.1.5.1. Nuevo Post

#### Descripción:

Un administrador crea un nuevo post, asignándole una categoría, y una fecha de publicación (opcional).

#### Precondición:

El usuario es administrador y tiene permisos suficientes.

#### Postcondición:

Existe el post creado con las características definidas.

El título del post es único.

El post pertenece a alguna categoría

#### Flujo básico:

1. El administrador redacta un nuevo post.
2. El administrador introduce los datos necesarios para crear el post.
3. Se validan los datos del post.
4. Si la categoría no existe se crea automáticamente.
5. Se guarda el nuevo post.

#### Flujo alternativo:

- 3.1 El título del post no es único.
- 3.2 Mensaje de error. Se solicita un nuevo título.
- 3.3 Se reintenta.

4.1 Error de conexión con la base de datos.

4.2 Mensaje de error.

4.3 Se detiene la ejecución.

5.1 Error de conexión con la base de datos.

5.2 Mensaje de error.

5.3 Se detiene la ejecución.

#### 2.1.5.2. Editar Post

##### **Descripción:**

Permite a un administrador realizar modificaciones sobre un post existente.

##### **Precondición:**

El usuario es administrador y tiene permisos suficientes.

El post objetivo existe.

##### **Postcondición:**

El post editado tiene las características definidas por el administrador.

El título del post es único.

El post pertenece a alguna categoría.

##### **Flujo básico:**

1. El administrador selecciona el post que quiere editar.
2. El administrador edita los datos necesarios que crea convenientes.
3. Se validan los datos.
4. Si la categoría no existe se crea automáticamente.
5. Se guarda el post.

##### **Flujo alternativo:**

- 3.1 El título del post no es único.
- 3.2 Mensaje de error. Se solicita un nuevo título.
- 3.3 Se reintenta.

4.1 Error de conexión con la base de datos.

4.2 Mensaje de error.

4.3 Se detiene la ejecución.

5.1 Error de conexión con la base de datos.

5.2 Mensaje de error.

5.3 Se detiene la ejecución.

El caso de uso publicar post es un caso especial de editar post, ya que la publicación de un post solo corresponde a la edición del campo que indica si el post ha sido publicado o no.

#### 2.1.5.3. Moderar comentario.

**Descripción:**

Valida un comentario como apto para su publicación, para evitar problemas.

**Precondición:**

El usuario es administrador y tiene permisos suficientes.

El comentario existe y pertenece a un post publicado.

**Postcondición:**

El comentario objetivo está moderado.

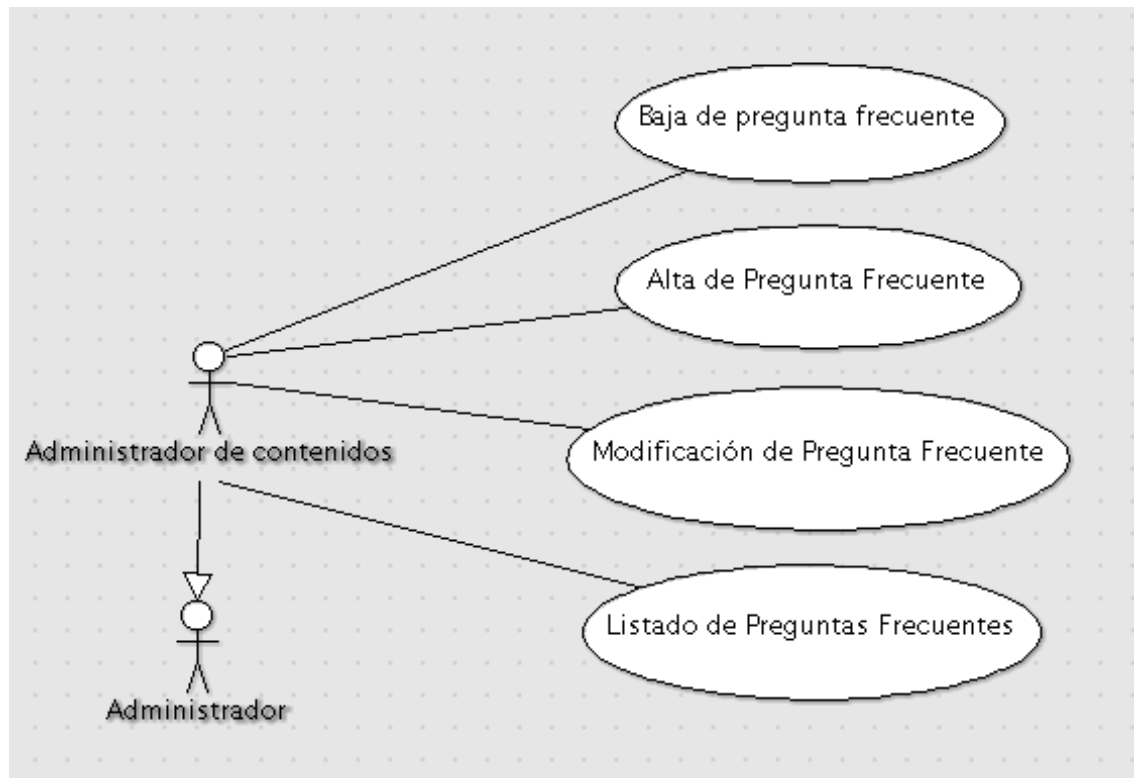
**Flujo básico:**

1. El administrador selecciona el comentario a moderar.
2. El administrador marca el comentario como válido o no.
3. Se actualiza el comentario objetivo.

**Flujo alternativo:**

- 1.1 Error de conexión con la base de datos.
- 1.2 Mensaje de error.
- 1.3 Se detiene la ejecución.
  
- 2.1 Error de conexión con la base de datos.
- 2.2 Mensaje de error.
- 2.3 Se detiene la ejecución.
  
- 3.1 Error de conexión con la base de datos.
- 3.2 Mensaje de error.
- 3.3 Se detiene la ejecución.

## 2.1.6. Gestor de Preguntas Frecuentes



El gestor de preguntas frecuentes y el gestor del glosario son prácticamente lo mismo, la única diferencia es que las preguntas frecuentes se agrupan en categorías. La sección de modificación debe permitir activar y desactivar preguntas sin borrarlas. Este gestor debe estar preparado teniendo en cuenta todos los idiomas disponibles.

### 2.1.6.1. Alta de pregunta frecuente

#### Descripción:

Un administrador crea una nueva FAQ, asignándole una categoría.

#### Precondición:

El usuario es administrador y tiene permisos suficientes.

#### Postcondición:

Existe la FAQ con las características definidas.

El título de la FAQ es único.

#### Flujo básico:

1. El administrador escoge crear una nueva FAQ.
2. El administrador introduce los datos necesarios para crear la FAQ.
3. Se validan los datos.
4. Si la categoría no existe se crea automáticamente.
5. Se guarda la nueva FAQ.

**Flujo alternativo:**

- 3.1 El título de la FAQ no es único.
- 3.2 Mensaje de error. Se solicita un nuevo título.
- 3.3 Se reintenta.

- 4.1 Error de conexión con la base de datos.
- 4.2 Mensaje de error.
- 4.3 Se detiene la ejecución.

- 5.1 Error de conexión con la base de datos.
- 5.2 Mensaje de error.
- 5.3 Se detiene la ejecución.

2.1.6.2. Baja de pregunta frecuente

**Descripción:**

Un administrador da de baja una FAQ.

**Precondición:**

El usuario es administrador y tiene permisos suficientes.  
La FAQ existe y está activa.

**Postcondición:**

La FAQ con las características definidas está marcada como inactiva (no existe la baja real).

**Flujo básico**

- 1. El administrador selecciona la FAQ a eliminar.
- 2. Se pide confirmación al usuario
- 3. Se marca como inactiva la FAQ seleccionada.

**Flujo alternativo**

- 3.1 Error de conexión con la base de datos.
- 3.2 Mensaje de error.
- 3.3 Se detiene la ejecución.

2.1.6.3. Modificación de pregunta frecuente

**Descripción:**

Un administrador modifica una FAQ.

**Precondición:**

El usuario es administrador y tiene permisos suficientes.

**Postcondición:**

Existe la FAQ con las nuevas características  
El título de la FAQ es único.

**Flujo básico:**

1. El administrador escoge la FAQ que quiere modificar.
2. El administrador introduce los datos necesarios para crear la FAQ.
3. Se validan los datos
4. Si la categoría no existe se crea automáticamente.
5. Se guarda la nueva FAQ.

**Flujo alternativo:**

- 3.1 El título de la FAQ no es único.
- 3.2 Mensaje de error. Se solicita un nuevo título.
- 3.3 Se reintenta.
  
- 4.1 Error de conexión con la base de datos.
- 4.2 Mensaje de error.
- 4.3 Se detiene la ejecución.
  
- 5.1 Error de conexión con la base de datos.
- 5.2 Mensaje de error.
- 5.3 Se detiene la ejecución.

2.1.6.4. Listado de preguntas frecuentes

**Descripción:**

Se muestran las FAQ atendiendo a criterios del usuario.

**Precondición:**

El usuario es administrador y tiene permisos suficientes.

**Postcondición:**

Se muestran las FAQ que cumplen las condiciones impuestas por el usuario.

**Flujo básico:**

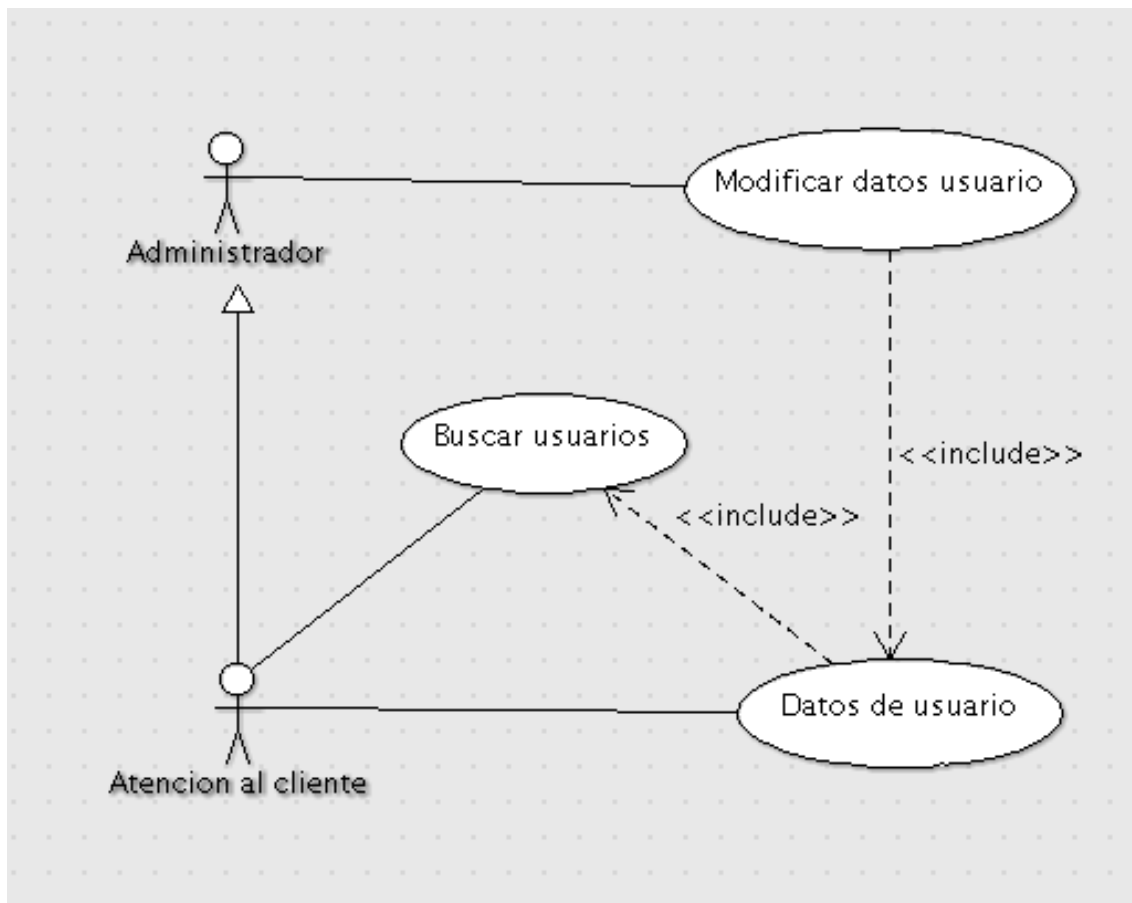
1. El administrador escoge el listado que quiere.
2. Se muestran las FAQ que verifiquen los requisitos.

**Flujo alternativo:**

- 2.1 Error de conexión con la base de datos.
- 2.2 Mensaje de error.
- 2.3 Se detiene la ejecución.



### 2.1.7. Gestión de usuarios



La gestión de usuarios permite consultar los datos de un usuario y modificarlos si es necesario. Dependiendo de que tipo de usuario administrador solo se mostraran algunos datos. Un usuario de atención al cliente no tiene acceso a la modificación manual de contraseñas... Si se modifican cambios sobre campos del juego, contraseña, clave de firma... se le deben notificar al usuario por correo electrónico de inmediato.

#### 2.1.7.1. Buscar usuarios

**Descripción:**

Permite búsquedas en los listados de usuarios según múltiples parámetros.

**Precondición:**

El usuario es administrador y tiene permisos suficientes.

**Postcondición:**

Se muestra un listado de usuarios que cumplen todas las condiciones especificadas en la búsqueda.

**Flujo básico:**

1. El administrador introduce los datos para la búsqueda.
2. Se muestran los usuarios que verifiquen los requisitos.

**Flujo alternativo:**

- 2.1 Error de conexión con la base de datos.
- 2.2 Mensaje de error.
- 2.3 Se detiene la ejecución.

2.1.7.2. Datos de usuario

**Descripción:**

Muestra los datos de un usuario concreto.

**Precondición:**

El usuario es administrador y tiene permisos suficientes.  
El usuario existe.

**Postcondición:**

Se muestran los datos del usuario seleccionado.

**Flujo básico:**

- 1. El administrador hace clic sobre un usuario.
- 2. Se muestran los resultados.

**Flujo alternativo:**

- 2.1 Error de conexión con la base de datos.
- 2.2 Mensaje de error.
- 2.3 Se detiene la ejecución.

2.1.7.3. Modificar datos de usuario

**Descripción:**

Muestra los datos de un usuario concreto y permite editar algunos.

**Precondición:**

El usuario es administrador y tiene permisos suficientes.  
El usuario existe.

**Postcondición:**

Se guarda el usuario objetivo con los cambios realizados.

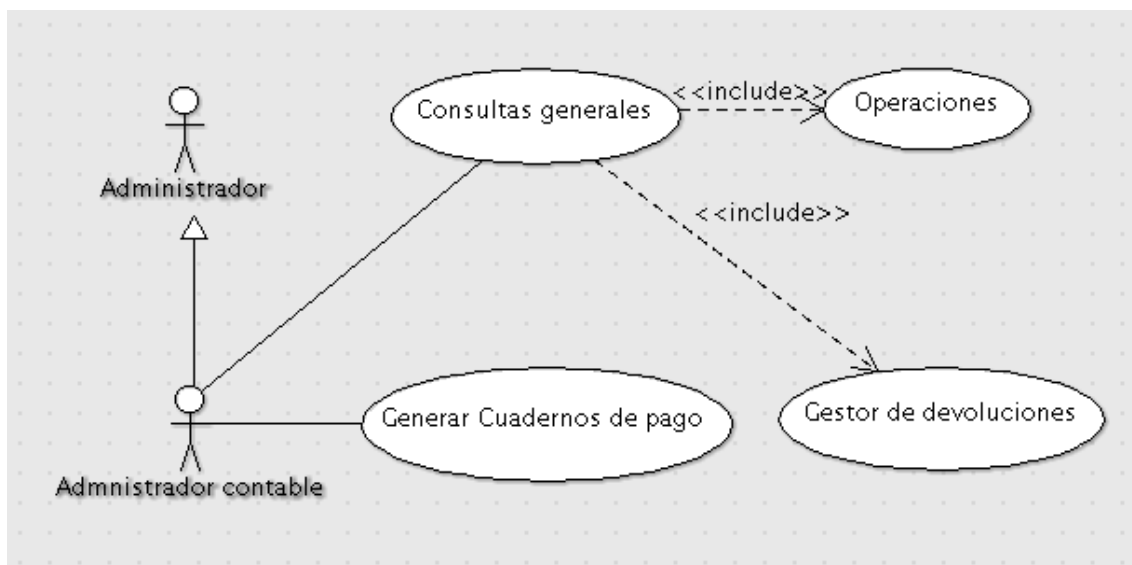
**Flujo básico:**

- 1 El administrador hace clic sobre un usuario.
- 2 Se muestran los resultados.
- 3 El administrador edita uno o más de los campos permitidos.
- 4 Se validan los datos.
- 5 Se guardan los cambios.
- 6 Se manda un correo al usuario si es necesario.

### Flujo alternativo:

- 2.1 Error de conexión con la base de datos.
- 2.2 Mensaje de error.
- 2.3 Se detiene la ejecución.
  
- 4.1 Los datos no son correctos
- 4.2 Mensaje de error. Se le pide al administrador que corrija los datos erróneos.
- 4.3 Se reintenta.
  
- 5.1 Error de conexión con la base de datos.
- 5.2 Mensaje de error.
- 5.3 Se detiene la ejecución.
  
- 6.1 Error al enviar el correo electrónico.
- 6.2 Mensaje de error.
- 6.3 No se realizan los cambios involucrados por el correo.

### 2.1.8. Contabilidad



La sección de contabilidad permite consultar las operaciones bancarias, realizadas durante un periodo determinado por el usuario, desglosadas según el IVA correspondiente en cada caso. Además el administrador contable puede añadir y eliminar devoluciones.

Otra de las funciones del administrador contable es realizar las transferencias de pago a los usuarios que solicitan cobrar su dinero. El administrador le proporcionará una herramienta que genera un fichero con formato estándar de cuaderno 34.1 a partir de las peticiones de cobro de los usuarios.

#### 2.1.8.1. Consultas generales

**Descripción:**

Muestra los datos contables de un periodo concreto: cobros, pagos, devoluciones... desglosados como corresponda.

**Precondición:**

El usuario es administrador y tiene permisos suficientes.

**Postcondición:**

Se muestran los datos del periodo seleccionado.

**Flujo básico:**

1. El administrador selecciona el periodo a consultar.
2. Se muestran los resultados contables de ese periodo.

**Flujo alternativo:**

- 1.1 El periodo está mal construido
- 1.2 Mensaje de error. Se le informa al usuario del problema.
- 1.3 Se reintenta
- 2.1 Error de conexión con la base de datos.
- 2.2 Mensaje de error.
- 2.3 Se detiene la ejecución.

2.1.8.2. Operaciones

**Descripción**

Muestra los pagos de usuarios a través del TPV dentro de un periodo concreto.

**Precondición**

El usuario es administrador y tiene permisos suficientes.  
El periodo a consultar está bien construido.

**Postcondición:**

Se muestran los datos del periodo seleccionado.

**Flujo básico:**

1. El administrador selecciona el periodo a consultar.
2. Se muestran los resultados contables de ese periodo.

**Flujo alternativo:**

- 1.1 El periodo está mal construido
- 1.2 Mensaje de error. Se le informa al usuario del problema.
- 1.3 Se reintenta
- 2.1 Error de conexión con la base de datos.
- 2.2 Mensaje de error.
- 2.3 Se detiene la ejecución.

2.1.8.3. Generar cuadernos de pago

**Descripción:**

Genera un listado de órdenes de transferencia en el formato 34.1.

**Precondición:**

El usuario es administrador y tiene permisos suficientes.

**Postcondición:**

El usuario ha descargado el fichero con todas las transferencias del día correspondiente con el formato adecuado.

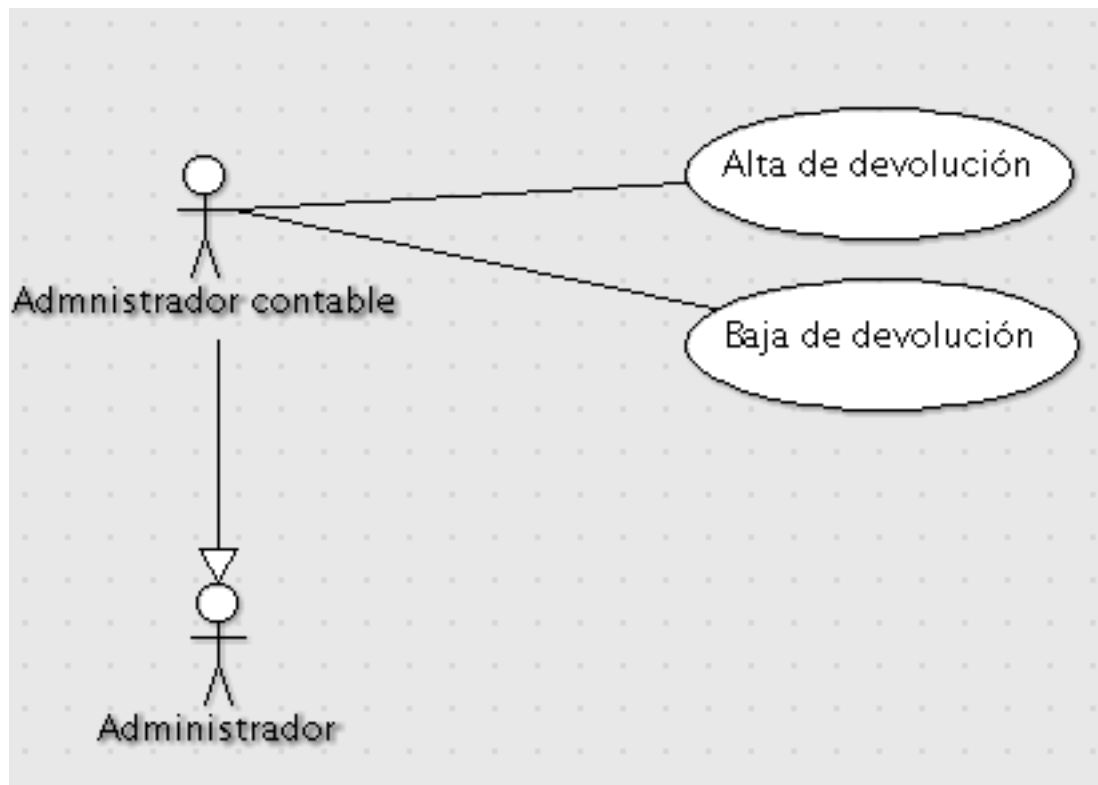
**Flujo básico:**

1. El administrador selecciona el día del que desea sacar el listado.
2. Se obtienen las peticiones de cobro del día correspondiente.
3. Se genera el fichero con las líneas generadas.
4. El usuario hace clic en descargar.
5. El usuario descarga el fichero.

**Flujo alternativo:**

- 2.1 Error de conexión con la base de datos.
- 2.2 Mensaje de error.
- 2.3 Se detiene la ejecución.

### 2.1.9. Gestor de devoluciones



Cada vez que se realice una devolución a un cliente es necesario darla de alta en el sistema para que las cuentas cuadren (las devoluciones como tales se realizan a través de la plataforma bancaria correspondiente).

#### 2.1.9.1. Alta de devolución

**Descripción:**

Se da de alta la devolución de una operación de cobro (la devolución real se realiza por los medios habituales).

**Precondición:**

El usuario es administrador y tiene permisos suficientes.  
La operación que se va a anotar existe.

**Postcondición:**

La devolución se ha creado correctamente.

**Flujo básico:**

1. El administrador selecciona la operación a anotar.
2. El administrador introduce la fecha de la devolución.
3. Se crea la devolución de esa operación.

**Flujo alternativo:**

- 3.1 Error de conexión con la base de datos.
- 3.2 Mensaje de error.
- 3.3 Se detiene la ejecución.

- 3.1 La operación ya ha sido devuelta
- 3.2 Se termina la ejecución

2.1.9.2. Baja de devolución

**Descripción:**

Se da de baja la (petición de) devolución de una operación de cobro.

**Precondición:**

El usuario es administrador y tiene permisos suficientes.  
La operación que se va a dar de baja existe.

**Postcondición:**

La devolución se ha eliminado correctamente.

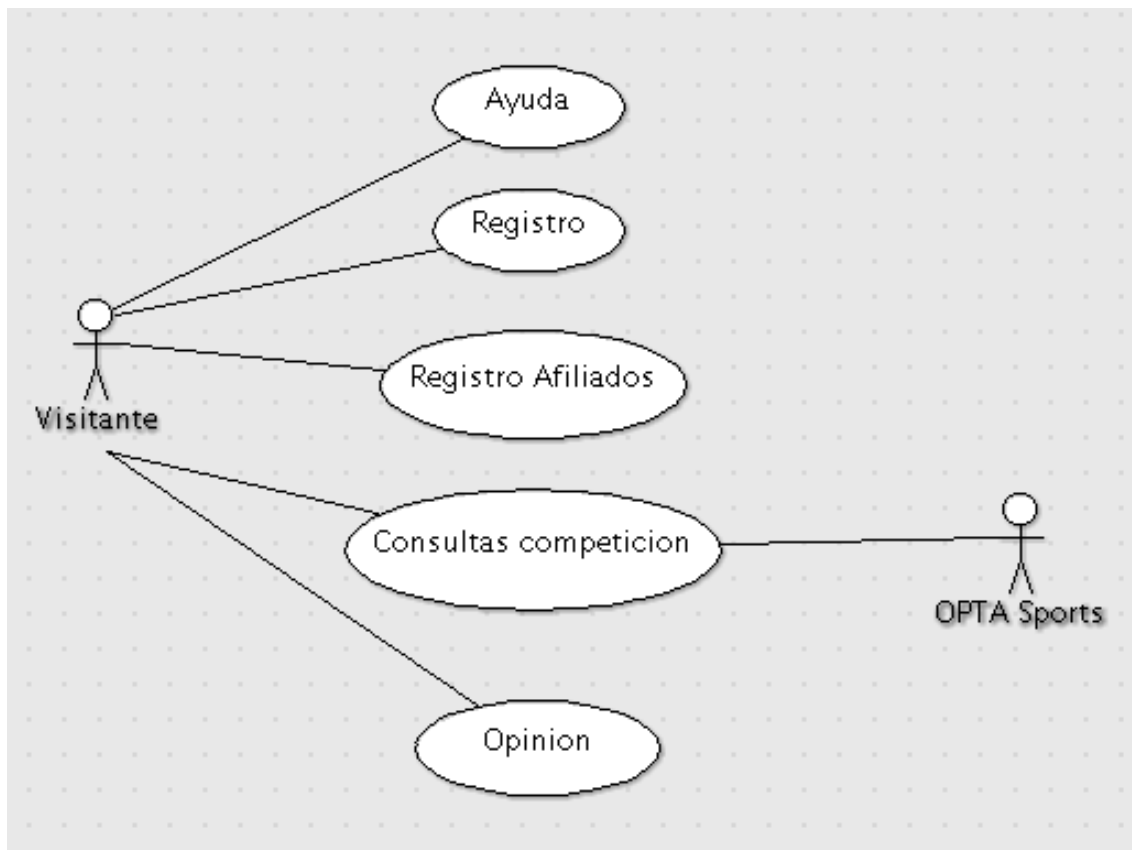
**Flujo básico:**

1. El administrador selecciona la devolución a eliminar.
2. Se crea la devolución de esa operación.

**Flujo alternativo:**

- 2.1 Error de conexión con la base de datos.
- 2.2 Mensaje de error.
- 2.3 Se detiene la ejecución.

### 2.1.10. Visitante



Un visitante sin registrar en la página puede realizar el proceso de registro para convertirse en usuario o afiliado. También puede navegar por las diferentes secciones de la página pero sin acceder a la zona de usuarios. También disponen de una zona donde consultar consejos y dejar los suyos propios.

#### 2.1.10.1. Registro

**Descripción:**

Permite registrarse como usuario a un visitante

**Precondición:**

No existe sesión de usuario

**Postcondición:**

Se ha enviado un correo al usuario.

Se ha registrado el envío en BD.

Se ha creado un nuevo registro de usuario en la BD.

El nombre de usuario es único.

**Flujo básico:**

1. El usuario introduce los datos de registro.
2. Se validan los datos de registro
3. Se envía un correo al usuario y se completa el registro.

**Flujo alternativo:**

- 2.1 Los datos de usuario no cumplen los requisitos.
- 2.2 Mensaje de error. Se le pide al usuario que complete los datos erróneos.
- 2.3 Se reintenta.

- 3.1 Error de conexión con la base de datos.
- 3.2 Mensaje de error
- 3.3 Finaliza la ejecución

- 3.1 Error al enviar el correo electrónico
- 3.2 No se envía. Mensaje de error.

2.1.10.2. Registro afiliados

**Descripción:**

Permite realizar una petición de afiliación a un visitante, esta petición deberá ser aceptada por un administrador del sitio.

**Precondición:**

No existe sesión de usuario.

**Postcondición:**

Se ha enviado un correo al usuario.

Se ha creado una nueva petición de afiliación en la base de datos.

**Flujo básico:**

- 1. El usuario introduce los datos de registro.
- 2. Se validan los datos de registro.
- 3. Se envía un correo al usuario y se completa el registro.

**Flujo alternativo:**

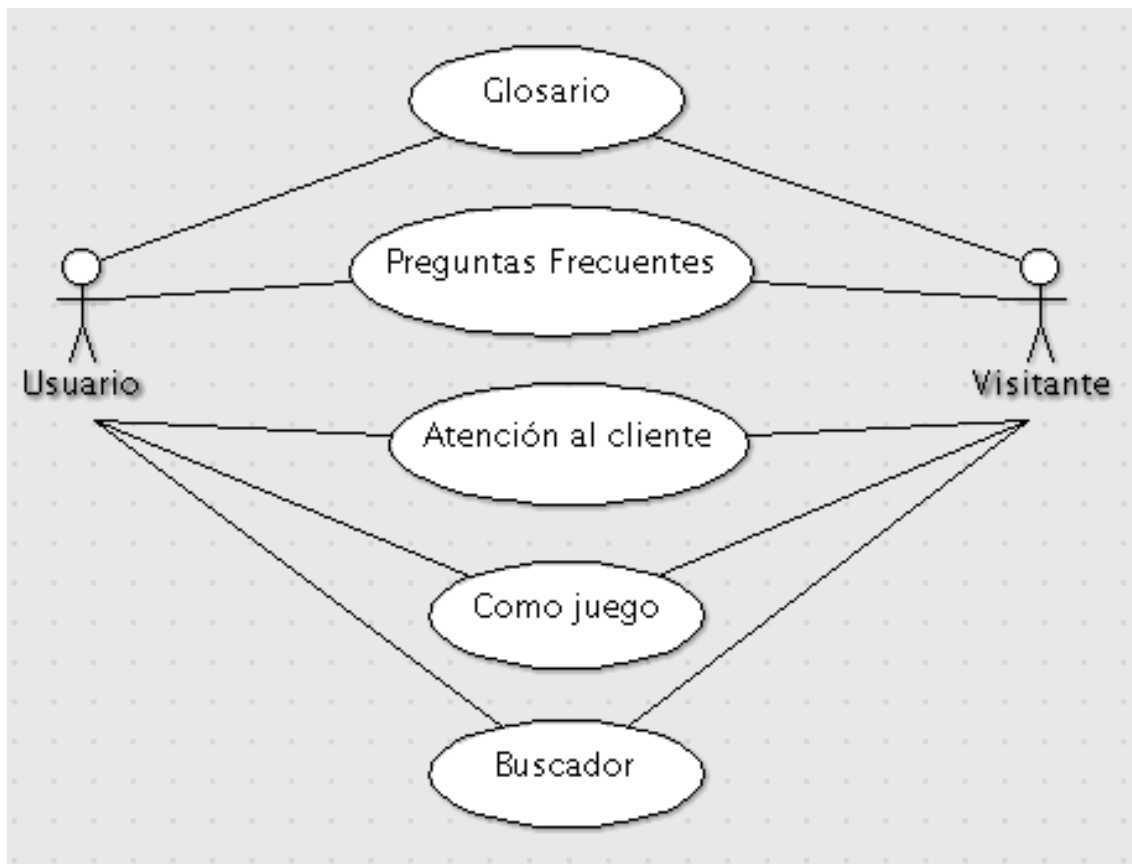
- 2.1 Los datos de usuario no cumplen los requisitos.
- 2.2 Mensaje de error. Se le pide al usuario que complete los datos erróneos.
- 2.3 Se reintenta.

- 3.1 Error de conexión con la base de datos.
- 3.2 Mensaje de error.
- 3.3 Finaliza la ejecución.

- 3.1 Error al enviar el correo electrónico.
- 3.2 No se envía. Mensaje de error.



### 2.1.11. Ayuda



La sección de ayuda contiene información sobre el juego organizada de diferentes maneras según la sección a la que se acceda. Existen un glosario de términos, con sus correspondientes definiciones; una sección de preguntas frecuentes, con ayuda en las cuestiones que más dudas presenten a los usuarios y una guía rápida del juego en la sección “como juego”. Además los usuarios tienen un buscador y una sección desde la que enviar sus dudas.

#### 2.1.11.1. Glosario

**Descripción:**

Muestra los términos del glosario de la letra seleccionada (por defecto la A)

**Precondición:**

Ninguna.

**Postcondición:**

Se muestran los datos del glosario por orden alfabético para la letra seleccionada.

**Flujo básico:**

1. El usuario selecciona la letra.
2. Se muestran los datos.

**Flujo alternativo:**

- 1.1 Los parámetros no son correctos.
- 1.2 Error 404.

- 2.1 Error de conexión con la base de datos.
- 2.2 Mensaje de error.
- 2.3 Finaliza la ejecución.

#### 2.1.11.2. Preguntas frecuentes

**Descripción:**

Muestra las FAQ agrupadas por categoría.

**Precondición:**

Ninguna.

**Postcondición:**

Se muestran las preguntas frecuentes de la categoría correspondiente.

**Flujo básico:**

1. El usuario selecciona la categoría.
2. Se muestran los datos.

**Flujo alternativo:**

- 1.1 Los parámetros no son correctos.
- 1.2 Error 404
- 2.1 Error de conexión con la base de datos.
- 2.2 Mensaje de error.
- 2.3 Finaliza la ejecución.

#### 2.1.11.3 Cómo juego

**Descripción:**

Muestra los conceptos básicos del juego, actúa como una guía rápida del mismo.

**Precondición:**

Ninguna.

**Postcondición:**

Se muestra la página.

**Flujo básico:**

Se muestra la página.

**Flujo alternativo:**

Ninguno.

#### 2.1.11.4. Buscador

**Descripción:**

Permite realizar búsquedas en el contenido de la página. El usuario puede buscar en toda la página o restringir su búsqueda a secciones predeterminadas.

**Precondición:**

Ninguna.

**Postcondición:**

Se muestran los resultados.

**Flujo básico:**

1. El usuario introduce la búsqueda y dónde desea buscar.
2. Se muestran los datos.

**Flujo alternativo:**

- 1.1 Los parámetros no son correctos.
- 1.2 Error 404
- 1.1 La búsqueda es maliciosa.
- 1.2 No se realiza la búsqueda.

#### 2.1.11.5. Atención al cliente

**Descripción:**

Permite enviar correos de atención al cliente.

**Precondición:**

Ninguna.

**Postcondición:**

Se envía un correo al buzón de atención al cliente.

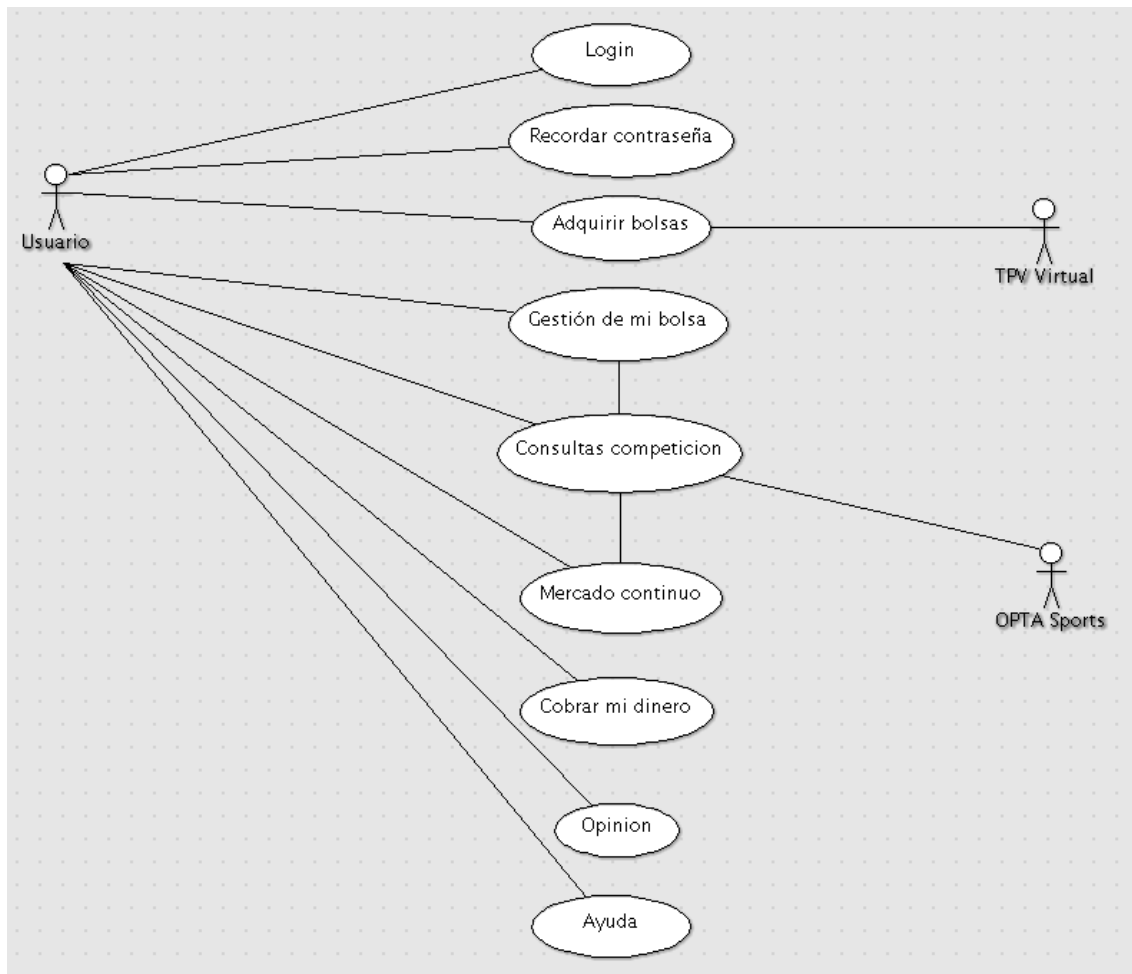
**Flujo básico:**

1. El usuario introduce los datos necesarios.
2. Se validan los datos.
3. Se envía un correo y se confirma el envío al usuario.

**Flujo alternativo:**

- 2.1 Los datos no cumplen los requisitos.
- 2.2 Mensaje de error. Se le pide al usuario que complete los datos erróneos.
- 2.3 Se reintenta.
- 3.1 Error al enviar el correo electrónico.
- 3.2 Mensaje de error.
- 3.3 Se reintenta.

## 2.1.12. Usuarios



Los usuarios son obviamente los actores principales del sistema. Para poder acceder al mismo tienen que logearse a través de su nombre de usuario y contraseña, si no la recuerdan, puede solicitarse una nueva, a través del formulario correspondiente. Los usuarios pueden añadir bolsas a su cuenta y gestionar la misma desde la sección Mi Bolsa. En cualquier momento un usuario puede solicitar la devolución de su dinero. Los usuarios pueden navegar por la página de la misma forma que haría un usuario normal.

La gestión de la bolsa y el mercado continuo, necesitan información relativa a la competición para mostrar los datos de la forma adecuada, ya que siempre que aparezca una tabla de equipos, estos se ordenaran según la clasificación actual.

### 2.1.12.1. Login

**Descripción:**

Permite al visitante ingresar a la aplicación como usuario.

**Precondición:**

Ninguna.

**Postcondición:**

El usuario tiene una sesión activa en el sistema.

**Flujo básico:**

1. El usuario introduce los datos de login.
2. Se validan.
3. El usuario es redirigido a la página principal.

**Flujo alternativo:**

- 1.1 Los datos de usuario no son correctos.
- 1.2 Se devuelve al usuario a la home con el mensaje correspondiente.
- 2.1 Error de conexión con la base de datos.
- 2.2 Mensaje de error.
- 2.3 Finaliza la ejecución

2.1.12.2. Recordar contraseña

**Descripción:**

Se envía un correo al usuario con una nueva contraseña.

**Precondición:**

No existe sesión de usuario.

**Postcondición:**

Se ha cambiado la contraseña al usuario

Se le ha enviado un correo con la nueva contraseña al usuario.

**Flujo básico:**

1. El usuario introduce los datos necesarios.
2. Se validan los datos.
3. Se cambia la contraseña
4. Se envía un correo al usuario.

**Flujo alternativo:**

- 2.1 Los datos de usuario no son validados correctamente.
- 2.2 Se le pide al usuario que vuelva a completar los datos.
- 2.3 Se reintenta
- 3.1 Error de conexión con la base de datos.
- 3.2 Mensaje de error.
- 3.3 Finaliza la ejecución.
- 4.1 Se produce un error al enviar el correo.
- 4.2 Se le pide al usuario que vuelva a completar los datos.

### 2.1.12.3. Adquirir bolsas

**Descripción:**

Adquisición de bolsas por parte de los usuarios mediante pago a través de la tarjeta de crédito.

**Precondición:**

El usuario tiene una sesión activa.

**Postcondición:**

Se han incrementado las bolsas del usuario en la cantidad solicitada.

Se ha enviado un comprobante electrónico al usuario por correo electrónico.

**Flujo básico:**

1. El usuario introduce los datos necesarios.
2. Se validan los datos.
3. Se registra la petición en BD.
4. Se realiza la petición al TPV.  
(Aquí el usuario interactúa con el sistema bancario externo a través de un iframe y podría interrumpir el proceso cerrando el mismo)
5. Se recibe la respuesta del TPV.
6. Se marca la petición como aceptada en DB.
7. Se incrementan las bolsas del usuario.
8. Se envía un correo al usuario.

**Flujo alternativo:**

- 2.1 Los datos no cumplen los requisitos.
- 2.2 Mensaje de error. Se le indica al usuario que datos corregir.
- 2.3 Se reintenta.
  
- 3.1 Error de conexión con la base de datos.
- 3.2 Mensaje de error.
- 3.3 Finaliza la ejecución.
  
- 5.1 El TPV no acepta los datos.
- 5.2 Se marca la petición como denegada.
- 5.3 Se informa al usuario del problema.
  
- 6.1 Error de conexión con la base de datos.
- 6.2 Mensaje de error.
- 6.3 Finaliza la ejecución.
  
- 7.1 Error de conexión con la base de datos.
- 7.2 Mensaje de error.
- 7.3 Finaliza la ejecución.
  
- 8.1 Se produce un error al enviar el correo.
- 8.2 Mensaje de error. Se reintenta el envío.

#### 2.1.12.4. Mercado Continuo

**Descripción:**

Ofrece datos sobre el mercado en el instante actual, ofertas pendientes de ejecutar, ofertas totales ejecutadas, cotizaciones...

**Precondición:**

El usuario tiene una sesión activa

**Postcondición:**

Se muestran los datos por pantalla.

**Flujo básico:**

1. Se calculan los datos.
2. Se muestran los datos por pantalla.

**Flujo alternativo:**

- 1.1 Error de conexión con la base de datos.
- 1.2 Mensaje de error.
- 1.3 Finaliza la ejecución.

#### 2.1.12.5. Cobrar mi dinero

**Descripción:**

El usuario realiza una petición para cobrar su dinero, incluye su número de cuenta, al finalizar se le envía un correo para confirmar que todo ha funcionado correctamente.

**Precondición:**

El usuario tiene una sesión activa.

El usuario tiene efectivo.

El usuario no tiene acciones o la temporada ya ha finalizado.

**Postcondición:**

El usuario no tiene efectivo.

El usuario no tiene ofertas pendientes.

El usuario ha registrado en BD la petición.

**Flujo básico:**

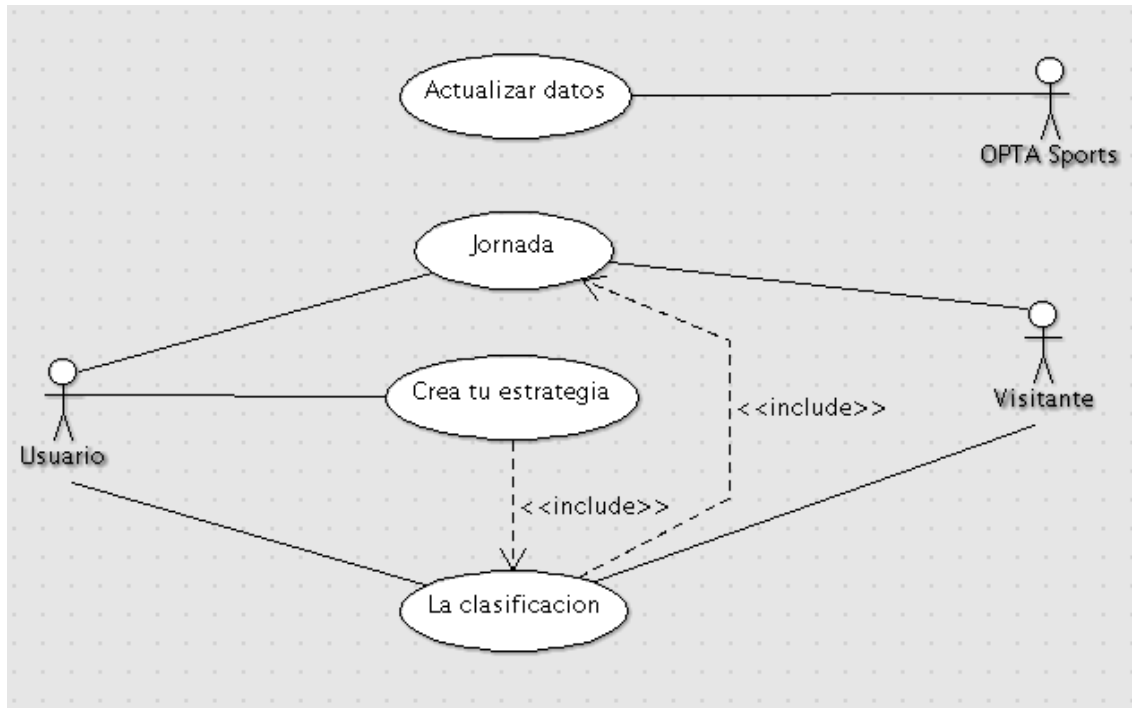
1. El usuario introduce los datos necesarios.
2. Se validan los datos.
3. Se registra la petición
4. Se envía un correo al usuario.

**Flujo alternativo:**

- 2.1 Los datos no cumplen los requisitos.
- 2.2 Mensaje de error. Se le indica al usuario que datos corregir.
- 2.3 Se reintenta.

- 3.1 Error de conexión con la base de datos.
- 3.2 Mensaje de error.
- 3.3 Finaliza la ejecución.
- 4.1 Se produce un error al enviar el correo.
- 4.2 Mensaje de error. Se reintentará el envío.

### 2.1.13. Consultas competición



Al ser un juego relacionado con el deporte resulta indispensable ofrecer información actualizada al momento sobre la competición que se está disputando. En nuestro caso, los datos que se ofrecen son, la clasificación e información sobre la jornada que se está disputando, a su vez cualquier usuario puede recorrer el calendario para consultar jornadas pasadas o futuras.

Los usuarios registrados disponen de una herramienta para simular clasificaciones según los resultados que creen que se pueden dar en las jornadas siguientes.

#### 2.1.13.1. La clasificación

##### Descripción:

Proporciona al usuario datos sobre la clasificación en el instante actual, el usuario puede avanzar o retrasar jornadas del calendario.

##### Precondición:

Ninguna.

##### Postcondición:

Se muestran los datos por pantalla.



**Flujo básico:**

1. Se calcula la jornada actual.
2. Se extrae la clasificación y la jornada actual más las dos jornadas siguientes.
3. Se muestran los datos por pantalla.

**Flujo alternativo:**

- 1.1 Error en los datos de competición.
- 1.2 Se emplean los últimos datos que funcionaron.

- 2.1 Error de conexión con la base de datos.
- 2.2 Mensaje de error.
- 2.3 Finaliza la ejecución.

- 2.1 Error en los datos de competición.
- 2.2 Se emplean los últimos datos que funcionaron.

2.1.13.2. La jornada

**Descripción:**

Datos sobre las jornadas en el instante actual (puede mostrar jornadas anteriores o futuras si lo solicita el usuario). Si la jornada se ha disputado incluye los resultados.

Este caso de uso se utiliza casi siempre como apoyo a otro tipo de datos o como complemento a otra información ofrecida.

**Precondición:**

El número de jornada esta dentro de los parámetros de la liga.

**Postcondición:**

Se muestran los datos por pantalla.

**Flujo básico:**

1. Se consulta la jornada solicitada.
2. Se devuelve un XML con los datos correspondientes.

**Flujo alternativo:**

- 1.1 Error en los datos de competición.
- 1.2 Se emplean los últimos datos que funcionaron.

- 2.1 Error de conexión con la base de datos.
- 2.2 Mensaje de error.
- 2.3 Finaliza la ejecución.

- 2.1 Error en los datos de competición.
- 2.2 Se emplean los últimos datos que funcionaron.

### 2.1.13.3. Crea tu estrategia

**Descripción:**

A partir de los datos sobre la clasificación en el instante actual, el usuario puede introducir los resultados esperados en la jornada o jornadas siguientes y calcular como quedaría la clasificación si se cumpliesen esos pronósticos.

**Precondición:**

El usuario tiene una sesión activa.  
La liga no ha terminado.

**Postcondición:**

Se muestra la clasificación según los datos que el usuario ha introducido.

**Flujo básico:**

1. Se calcula la jornada actual.
2. Se extrae la clasificación y la jornada actual más la jornada siguiente.
3. El usuario introduce los resultados (1X2) que espera durante todas las jornadas que quiera.
4. Se calcula la nueva clasificación.
5. Se muestran los datos por pantalla.

**Flujo alternativo:**

- 1.1 Error en los datos de competición.
- 1.2 Se emplean los últimos datos que funcionaron.
- 2.1 Error de conexión con la base de datos.
- 2.2 Mensaje de error.
- 2.3 Finaliza la ejecución.
- 2.1 Error en los datos de competición.
- 2.2 Se emplean los últimos datos que funcionaron.
- 3.1 Los datos no cumplen los requisitos.
- 3.2 Mensaje de error. Se le indica al usuario que datos corregir.
- 3.3 Se reintenta.
- 4.1 Error en el cálculo de la nueva clasificación.
- 4.2 Mensaje de error.
- 4.3 Finaliza la ejecución.

### 2.1.13.4. Actualizar datos

**Descripción:**

Cuando se produzcan cambios en la clasificación o en resultados estos cambios tienen que quedar reflejados en la aplicación. Una empresa externa debe proveernos y “darnos parte” cada vez que se producen estos cambios.

**Precondición:**

El XML que contiene los datos está bien formado.

**Postcondición:**

Los datos corresponden con el ultimo “informe” externo.

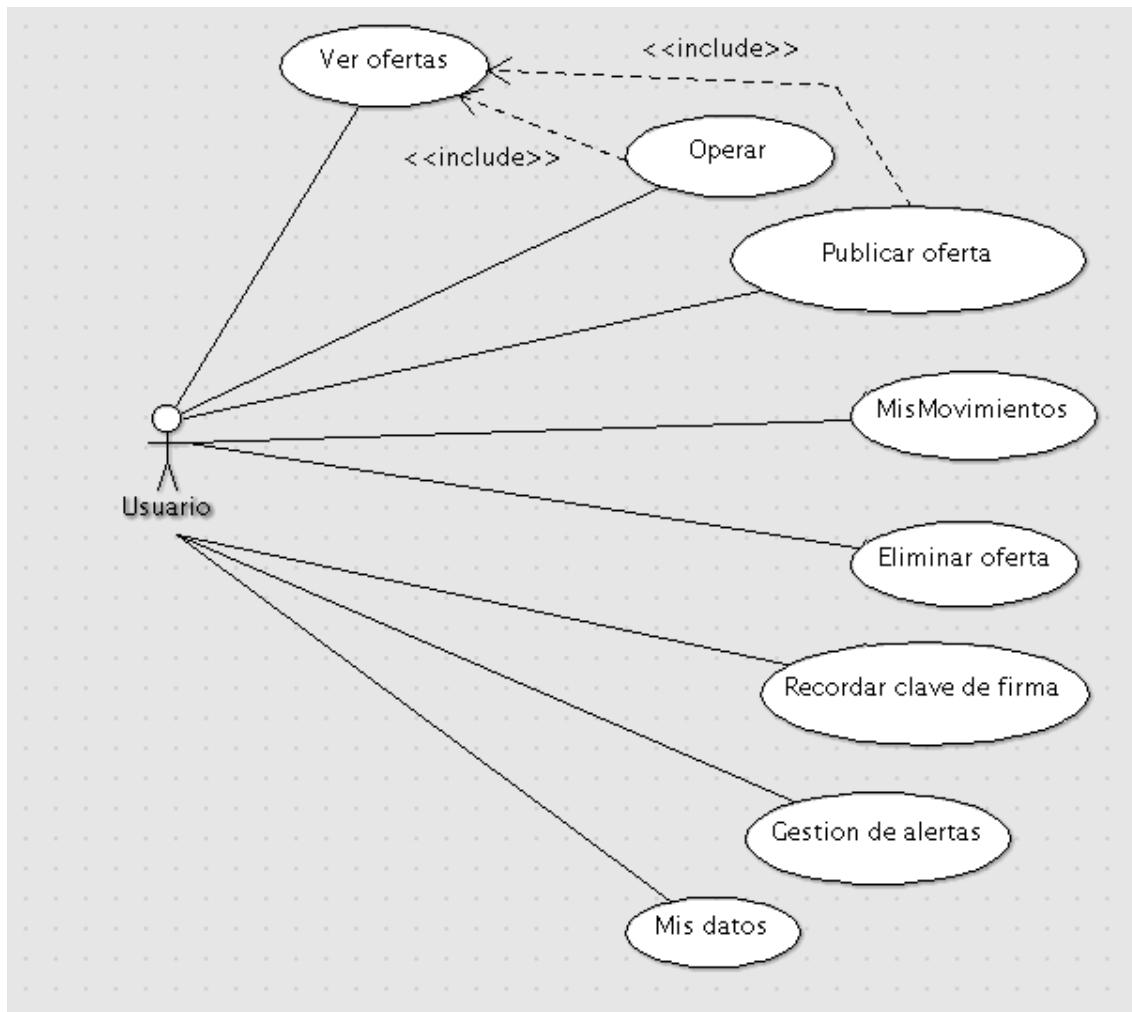
**Flujo básico:**

1. La empresa manda un XML con los datos.
2. Se procesa el XML.
3. Se guarda el contenido en BD.
4. Se actualizan los datos.

**Flujo alternativo:**

- 1.1 Error al recibir el fichero.
- 1.2 Se desecha y no se procesa.
- 1.3 Se solicita el reenvío del fichero.
  
- 2.1 Error al procesar el fichero.
- 2.2 Se desecha y no se procesa.
  
- 3.1 Error de conexión con la base de datos.
- 3.2 Mensaje de error.
- 3.3 Finaliza la ejecución.

## 2.1.14. Gestión de mi bolsa



Esta es la sección principal del juego ya que engloba las operaciones más importantes y es donde se desarrolla el mismo. Los usuarios aquí pueden consultar las ofertas publicadas por otros usuarios, si encuentran algo que les convence ejecutar la oferta correspondiente y si no, publicar una ellos mismos.

Además pueden consultar los movimientos que han realizado hasta el momento actual, configurar alertas de correo ante determinados eventos que pueden resultar interesantes. También pueden consultar y editar datos relacionados con su cuenta.

### 2.1.14.1. Ver ofertas

#### Descripción:

Se le muestran al usuario las 5 mejores ofertas para el equipo que seleccione. Si quiere operar solo podrá hacerlo sobre la mejor.

#### Precondición:

El usuario tiene una sesión activa.

La liga no ha terminado.

**Postcondición:**

Se muestran las 5 mejores ofertas agrupadas por precio ordenadas de más a menos beneficioso para el usuario, para ofertas de un mismo precio tienen preferencia las ofertas más antiguas. El usuario nunca debe saber contra quién está operando.

**Flujo básico:**

1. El usuario hace clic sobre comprar o vender acciones de un equipo.
2. Se obtienen las ofertas disponibles para ese equipo y la operación solicitada.
3. Se le muestran las ofertas al usuario.
4. Se completa el formulario con el número máximo de acciones que se pueden incluir en la operación.
- 4b. Si no hay ofertas se le muestra al usuario el mensaje correspondiente.
- 4c. Si el usuario no tiene dinero o acciones suficientes para operar se le muestra el mensaje correspondiente.

**Flujo alternativo:**

- 2.1 Error de conexión con la base de datos.
- 2.2 Mensaje de error.
- 2.3 Finaliza la ejecución.
  
- 3.1 No hay ofertas disponibles
- 3.2 Mensaje al usuario. Se le ofrece publicar una oferta.
  
- Si la operación es de venta.
- 4.1 El usuario no tiene acciones disponibles para vender.
- 4.2 Mensaje de error correspondiente.
  
- Si la operación es de compra.
- 4.1 El usuario no tiene efectivo disponible para comprar.
- 4.2 Mensaje al usuario. Se le ofrece publicar una oferta.

2.1.14.2. Operar

**Descripción:**

El usuario ejecuta alguna oferta disponible, es decir, compra o vende acciones directamente ejecutando la oferta que había hecho previamente otro usuario. Hay que tener en cuenta.

**Precondición:**

El usuario tiene una sesión activa.  
La liga no ha terminado.  
El usuario comprador tiene suficiente efectivo y el vendedor suficientes acciones.

**Postcondición:**

Las acciones y el efectivo de los usuarios involucrados en la transacción han sufrido las modificaciones necesarias.  
Se ha registrado la compraventa en BD.

**Flujo básico:**

1. El usuario introduce la cantidad que quiere comprar.
2. El usuario introduce su clave de firma.
3. Se guardan los históricos correspondientes.
4. Se efectúa la transacción.
  - 4.1 Se actualizan las acciones del usuario o usuarios que publicaron la oferta.
  - 4.2 Se actualiza el efectivo del usuario o usuarios que publicaron la oferta.
  - 4.3 Se actualizan las acciones del usuario que ejecuta la compra.
  - 4.4 Se actualiza el efectivo del usuario que ejecuta.
  - 4.5 Se borran las ofertas involucradas.
5. Se muestran las ofertas de nuevo.

**Flujo alternativo:**

- 2.1 La clave de firma no es correcta
- 2.2 Mensaje de error. Se le pide al usuario la clave de firma.
- 3.1 Error de conexión con la base de datos.
- 3.2 Mensaje de error.
- 3.3 Rollback de la transacción.
- 3.4 Finaliza la ejecución.
- 4.1 Error de conexión con la base de datos.
- 4.2 Mensaje de error.
- 4.3 Rollback de la transacción.
- 4.4 Finaliza la ejecución.
- 4.1 Error de coherencia de la transacción, los valores no son correctos.
- 4.2 Mensaje de error.
- 4.3 Rollback de la transacción.
- 4.4 Finaliza la ejecución.

2.1.14.3. Publicar ofertas

**Descripción:**

El usuario publica una oferta

**Precondición:**

El usuario tiene una sesión activa.

La liga no ha terminado.

El usuario tiene efectivo suficiente si la oferta es de compra.

El usuario tiene acciones suficientes si la oferta es de venta.

**Postcondición:**

La oferta está publicada.

**Flujo básico:**

1. El usuario hace clic en publicar oferta.
2. El usuario introduce los valores.
3. Se comprueba que la oferta sea mejor (para el que publica) que las existentes.
4. El usuario introduce la clave de firma.
5. Se registra la oferta en BD.

**Flujo alternativo:**

- 4.1 La clave de firma no es correcta.
- 4.2 Mensaje de error. Se le pide al usuario la clave de firma.
- 5.1 Error de conexión con la base de datos.
- 5.2 Mensaje de error.
- 5.3 Finaliza la ejecución.

2.1.14.4. Mis movimientos

**Descripción:**

Datos sobre los movimientos del usuario.

**Precondición:**

El usuario tiene una sesión activa.

**Postcondición:**

Se muestran los datos por pantalla.

**Flujo básico:**

- 1. El usuario hace clic sobre el tipo de movimientos que quiere ver.
- 2. Se obtienen los datos necesarios.
- 3. Se muestran los datos por pantalla.

**Flujo alternativo:**

- 2.1 Error de conexión con la base de datos.
- 2.2 Mensaje de error.
- 2.3 Finaliza la ejecución.

2.1.14.5. Eliminar ofertas

**Descripción:**

El usuario puede eliminar una oferta.

**Precondición:**

El usuario tiene una sesión activa.  
La oferta está pendiente de ejecutar.  
El usuario está consultando sus movimientos pendientes.

**Postcondición:**

La oferta se ha borrado de la base de datos.

**Flujo básico:**

- 1. El usuario hace clic en eliminar oferta, sobre la oferta que quiere eliminar.
- 2. El usuario introduce la clave de firma.
- 3. Se elimina la oferta en la base de datos.

**Flujo alternativo:**

- 2.1 La clave de firma no es correcta.
- 2.2 Mensaje de error. Se le pide al usuario la clave de firma.

- 3.1 Error de conexión con la base de datos.
- 3.2 Mensaje de error.
- 3.3 Finaliza la ejecución.

2.1.14.6. Recordar clave de firma

**Descripción:**

Se envía un correo al usuario con una nueva clave de firma.

**Precondición:**

El usuario tiene una sesión activa.

**Postcondición:**

Se ha cambiado la clave de firma al usuario.  
Se le ha enviado un correo con la nueva clave al usuario.  
Se ha registrado el envío en la base de datos.

**Flujo básico:**

- 1. El usuario introduce los datos necesarios.
- 2. Se validan los datos.
- 3. Se cambia la clave de firma.
- 4. Se envía un correo al usuario.

**Flujo alternativo:**

- 2.1 Los datos de usuario no son validados correctamente.
- 2.2 Se le pide al usuario que vuelva a completar los datos.
- 2.3 Se reintenta.

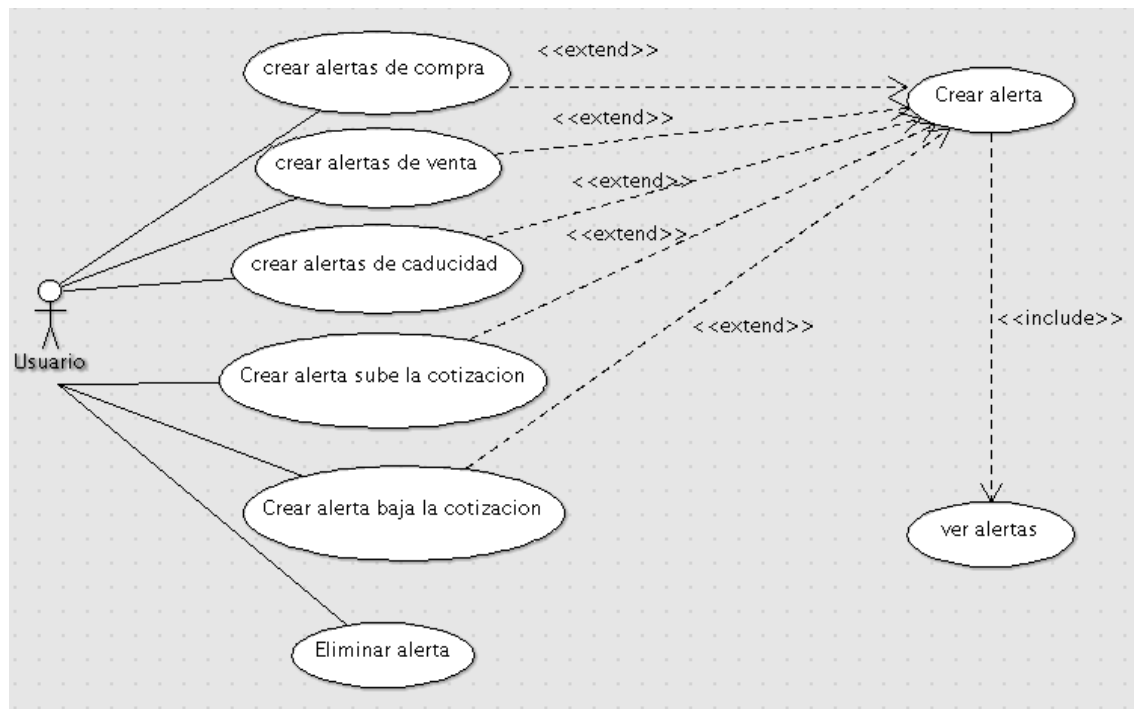
- 3.1 Error de conexión con la base de datos.
- 3.2 Mensaje de error.
- 3.3 Finaliza la ejecución.

- 4.1 Se produce un error al enviar el correo.
- 4.2 Se le pide al usuario que vuelva a completar los datos.

Los casos de uso Gestión de alertas y mis datos aún pueden refinarse más, por lo que sus cuadros resumen se desarrollarán más adelante.



### 2.1.15. Gestión de alertas



El usuario puede configurarse alertas de los eventos que se muestran en el diagrama para que le llegue un correo electrónico cuando estos se produzcan.

Las alertas de compra y venta envían un correo cuando le compran o venden acciones a un usuario. Por defecto están activadas para todos los equipos.

Las alertas de caducidad se envían cuando caduca una oferta sin que haya sido ejecutada por otro usuario.

Las alertas de sube o baja la cotización se envían cuando la cotización del equipo seleccionado sube o baja, se envían sólo una vez antes de ser eliminadas.

#### 2.1.15.1. Ver alertas

##### Descripción:

Muestra las alertas del usuario.

##### Precondición:

El usuario tiene una sesión activa.

##### Postcondición:

Se muestran los datos correctos.

##### Flujo básico:

1. El usuario entra la gestión de alertas.
2. Se le muestran los datos necesarios.

##### Flujo alternativo:

Hay un error al recuperar los datos.

Se muestra la página sin datos.

#### 2.1.15.2. Crear alerta

**Descripción:**

Crea una alerta del tipo correspondiente en la base de datos.

**Precondición:**

El usuario tiene una sesión activa.

**Postcondición:**

Se crea la alerta en la base de datos.

**Flujo básico:**

1. El usuario elige el tipo de alerta que quiere crear.
2. El usuario introduce los datos necesarios.
3. El usuario introduce la clave de firma.
4. Se crea la alerta en la base de datos.

**Flujo alternativo:**

- 2.1 Los datos no cumplen los requisitos.
- 2.2 Mensaje de error. Se le indica al usuario que datos corregir.
- 2.3 Se reintenta.
- 3.1 La clave de firma no es correcta.
- 3.2 Mensaje de error. Se le pide al usuario la clave de firma.
- 4.1 Error de conexión con la base de datos.
- 4.2 Mensaje de error.
- 4.3 Finaliza la ejecución.

Los diferentes casos de uso de crear una alerta de cada tipo son básicamente lo mismo que el caso de uso crear alerta, la única diferencia es la tabla donde se almacenan y la cantidad de datos necesarios. Por lo tanto no se realizará cuadro resumen sobre los mismos.

#### 2.1.15.3. Eliminar alerta

**Descripción:**

El usuario elimina una alerta existente.

**Precondición:**

El usuario tiene una sesión activa.  
La alerta existe.  
El usuario está en la gestión de alertas.

**Postcondición:**

La alerta se ha borrado de la BD.

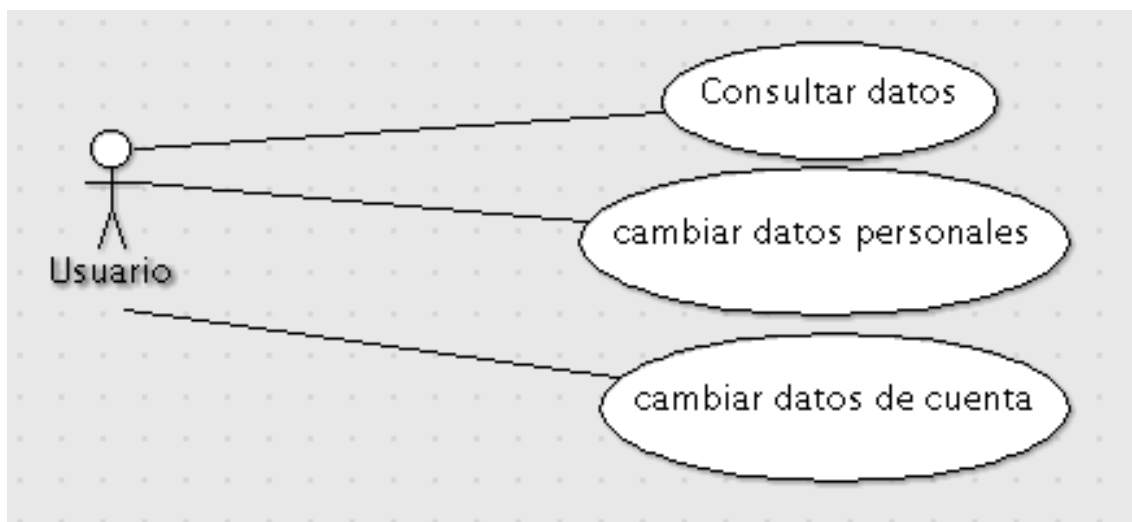
**Flujo básico:**

1. El usuario hace clic en eliminar alerta.
2. El usuario introduce la clave de firma.
3. Se elimina la alerta en BD.

**Flujo alternativo:**

- 2.1 La clave de firma no es correcta.
- 2.2 Mensaje de error. Se le pide al usuario la clave de firma.
- 3.1 Error de conexión con la base de datos.
- 3.2 Mensaje de error.
- 3.3 Finaliza la ejecución.

**2.1.16. Mis datos**



Los usuarios pueden consultar y editar datos relacionados con su cuenta: correo electrónico, liga de inicio preferida...

**2.1.16.1. Mis datos**

**Descripción:**

Muestra los datos del usuario

**Precondición:**

El usuario tiene una sesión activa.

**Postcondición:**

Se muestran los datos correctos.

**Flujo básico:**

1. El usuario entra la sección “Mis Datos”.
2. Se le muestran los datos pertinentes.

**Flujo alternativo:**

- 2.1 Error de conexión con la base de datos.
- 2.2 Mensaje de error.
- 2.3 Finaliza la ejecución.

2.1.16.2. Cambiar datos personales

**Descripción:**

Cambia los datos personales del usuario.

**Precondición:**

El usuario tiene una sesión activa.

**Postcondición:**

Se realizan los cambios solicitados.

**Flujo básico:**

1. El usuario realiza los cambios que considere oportunos sobre sus datos.
2. Los datos se actualizan en la base de datos.
3. Los datos se actualizan en la página del usuario.

**Flujo alternativo:**

- 2.1 Los datos no cumplen los requisitos.
  - 2.2 Mensaje de error. Se le indica al usuario que datos corregir.
  - 2.3 Se reintenta.
- 
- 2.1 Error de conexión con la base de datos.
  - 2.2 Mensaje de error.
  - 2.3 Finaliza la ejecución.

2.1.16.3. Cambiar datos de cuenta

**Descripción:**

Cambia los datos de la cuenta del usuario.

**Precondición:**

El usuario tiene una sesión activa.

**Postcondición:**

Se realizan los cambios solicitados.

**Flujo básico:**

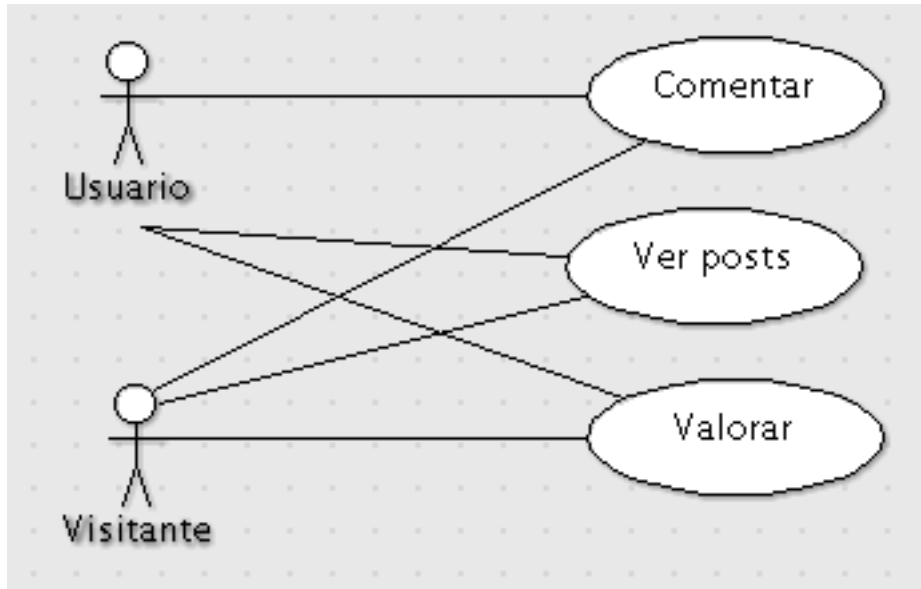
1. El usuario realiza los cambios que considere oportunos sobre sus datos.
2. Los datos se actualizan en la base de datos.
3. Los datos se actualizan en la página del usuario.

**Flujo alternativo:**

- 2.1 Los datos no cumplen los requisitos.
- 2.2 Mensaje de error. Se le indica al usuario que datos corregir.
- 2.3 Se reintenta.

- 2.1 Error de conexión con la base de datos.
- 2.2 Mensaje de error.
- 2.3 Finaliza la ejecución.

### 2.1.17. Opinión



Los usuarios pueden ver artículos de opinión y consejos sobre el juego, dejar los suyos propios, y puntuar positiva o negativamente cada artículo.

#### 2.1.17.1. Comentar

##### Descripción:

Permite comentar un post del blog. Los comentarios no aparecen publicados hasta que no son moderados por un administrador de contenidos.

##### Precondición:

El usuario se encuentra leyendo un post del blog.

##### Postcondición:

Se ha creado un comentario referente al post actual, que aún está sin moderar.

##### Flujo básico:

1. El usuario realiza el comentario e introduce los datos necesarios.
2. Se registra el comentario en la base de datos.

##### Flujo alternativo:

- 1.1 Los datos no cumplen los requisitos.
  - 1.2 Mensaje de error. Se le indica al usuario que datos corregir.
  - 1.3 Se reintenta.
- 
- 2.1 Error de conexión con la base de datos.
  - 2.2 Mensaje de error.
  - 2.3 Finaliza la ejecución.

#### 2.1.17.2. Valorar

**Descripción:**

Permite valorar (positivo o negativo) un post del blog.

**Precondición:**

El usuario se encuentra leyendo un post del blog.  
El usuario no ha valorado el post en esta sesión.

**Postcondición:**

Se cuenta el voto en el post correspondiente.

**Flujo básico:**

1. El usuario valora el post (clic en el botón correspondiente).
2. Se registra el voto en la base de datos.

**Flujo alternativo:**

- 2.1 Error de conexión con la base de datos.
- 2.2 Mensaje de error.
- 2.3 Finaliza la ejecución.

#### 2.1.17.3 Consultar datos

**Descripción:**

Muestra las cabeceras de los posts solicitados agrupados por categorías.

**Precondición:**

Ninguna.

**Postcondición:**

Se muestran los posts pertinentes.

**Flujo básico:**

1. El usuario solicita una categoría de posts.
2. Se le muestran los posts pertinentes.

**Flujo alternativo:**

- 1.1 La categoría no existe.
- 1.2 Error 404.
- 2.1 Error de conexión con la base de datos.
- 2.2 Mensaje de error.
- 2.3 Finaliza la ejecución.

### 3. Diseño

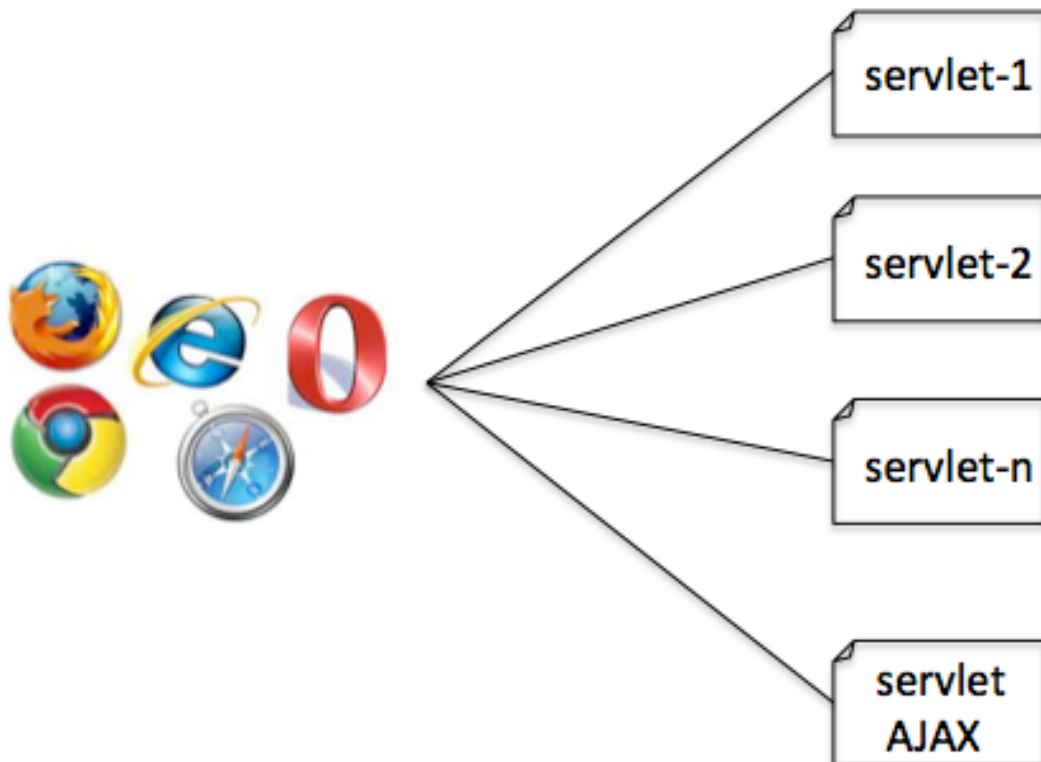
En este capítulo se tratarán los aspectos más interesantes del diseño del sistema.

#### 3.1 Funcionamiento general

Las peticiones realizadas a los servidores son atendidas por diferentes servlets que discriminan la petición a partir de la URL solicitada. A su vez, cada servlet tiene una serie de “acciones” asignadas de manera que, cuando llega la petición, ésta es tratada por la acción correspondiente. La mayor parte de los casos de uso tienen asignada una acción. Por ejemplo, una petición de la clasificación de la parte pública, sería atendida por el servlet público que, a su vez, llamaría a la acción correspondiente que sería la encargada de tratar y devolver los datos correspondientes.

A continuación se estudia el comportamiento del sistema con mayor profundidad:

Las peticiones llegan y son atendidas por el servidor Apache, que las traduce a partir del módulo de rewrite y las redirige al servidor Tomcat a través de un proxy. Tomcat discrimina la URL y atiende la petición con el servlet correspondiente.



Tomcat, junto con velocity, tratan la petición y construyen las plantillas que se enviarán al cliente.

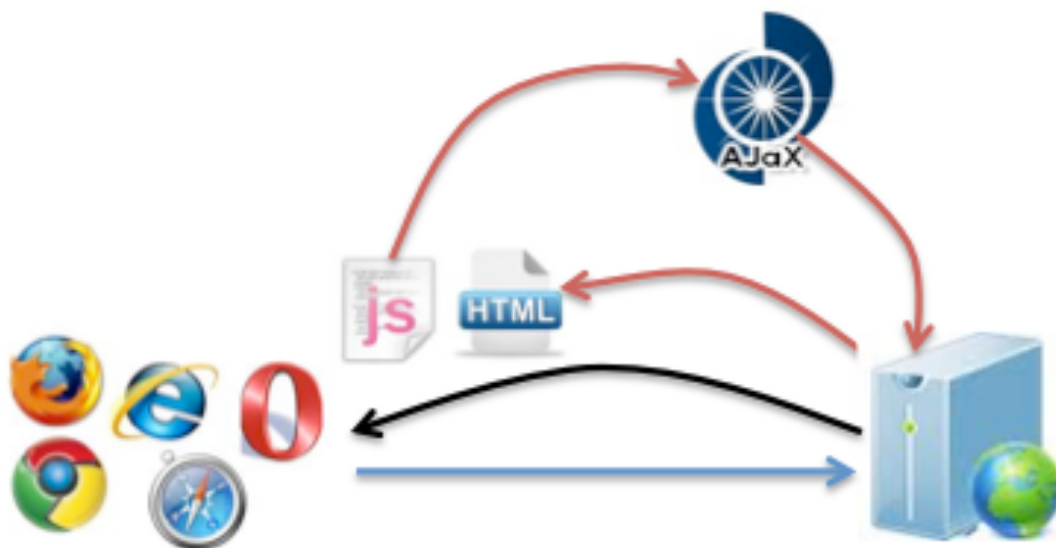


Este mecanismo de funcionamiento implica que el contenido público del sitio, no pueda ser “cacheado” por el servidor web y deba ser generado de forma dinámica obligatoriamente para cada visitante al portal.

Para poner en situación de lo que esto implica, supóngase que dos usuarios se conectan al sitio. Lo que sucederá es que se ejecutará el mismo JSP de inicio dos veces (uno para cada usuario) con independencia del idioma elegido. Es decir, si ambos eligen castellano (idioma por defecto del portal) se ejecutará dos veces obteniendo para ambos casos, como no podría ser de otra manera, el mismo resultado.

La consecuencia, el tiempo de presentación de las páginas públicas lleva una sobrecarga adicional que limitará de alguna manera el tiempo de respuesta del sitio en su conjunto, ralentizando en cierta medida a la parte realmente dinámica del juego.

Una vez el cliente ha recibido la plantilla, si es necesario mantener la información de la misma actualizada, el código JavaScript se encarga de realizar peticiones periódicas de información para actualizar los campos necesarios



Se están usando componentes AJAX para generación de HTML/JavaScript. El uso de este tipo de componentes en los documentos HTML permite que se pueda interactuar de forma dinámica con el usuario sin la obligación de tener que generar o recargar un nuevo documento HTML.



En el método tradicional de funcionamiento de HTML, cuando el usuario realiza una solicitud de información al servidor WEB, envía un formulario de datos mediante un comando POST. La respuesta del servidor a esta petición es enviar una nueva página HTML completa que debe ser repintada en el navegador.

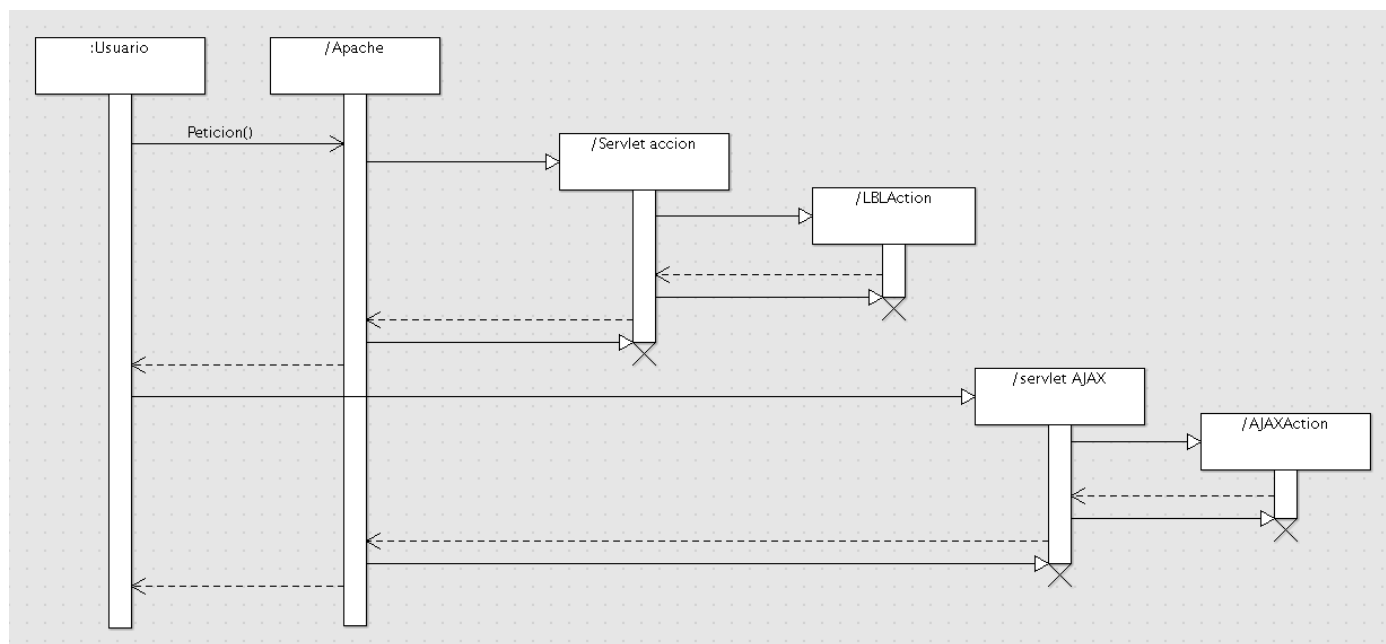
Por el contrario, el uso de JavaScript AJAX permite que la respuesta del servidor sea un pequeño documento XML que, al ser interpretado tras su recepción por el script, inserte en la plantilla únicamente la información recibida. Esto reduce dos cosas, el ancho de banda necesario para presentar la información y el tiempo que el navegador utilizará para “recargar” los datos recibidos.

Esta técnica se utilizará, por ejemplo, para recibir la información de las cotizaciones de las acciones del juego o la información asociada a la petición de órdenes de compraventa.

Esto implica que parte de la lógica de negocio será ejecutada en estos pequeños componentes (scripts) que corren en los navegadores de los usuarios.

Como se ha comentado anteriormente, las ventajas de AJAX son importantes, pero impone la obligatoriedad de que los navegadores tengan habilitado el uso de JavaScript para poder acceder a la funcionalidad completa del juego.

Un esquema general del funcionamiento descrito podría ser este:

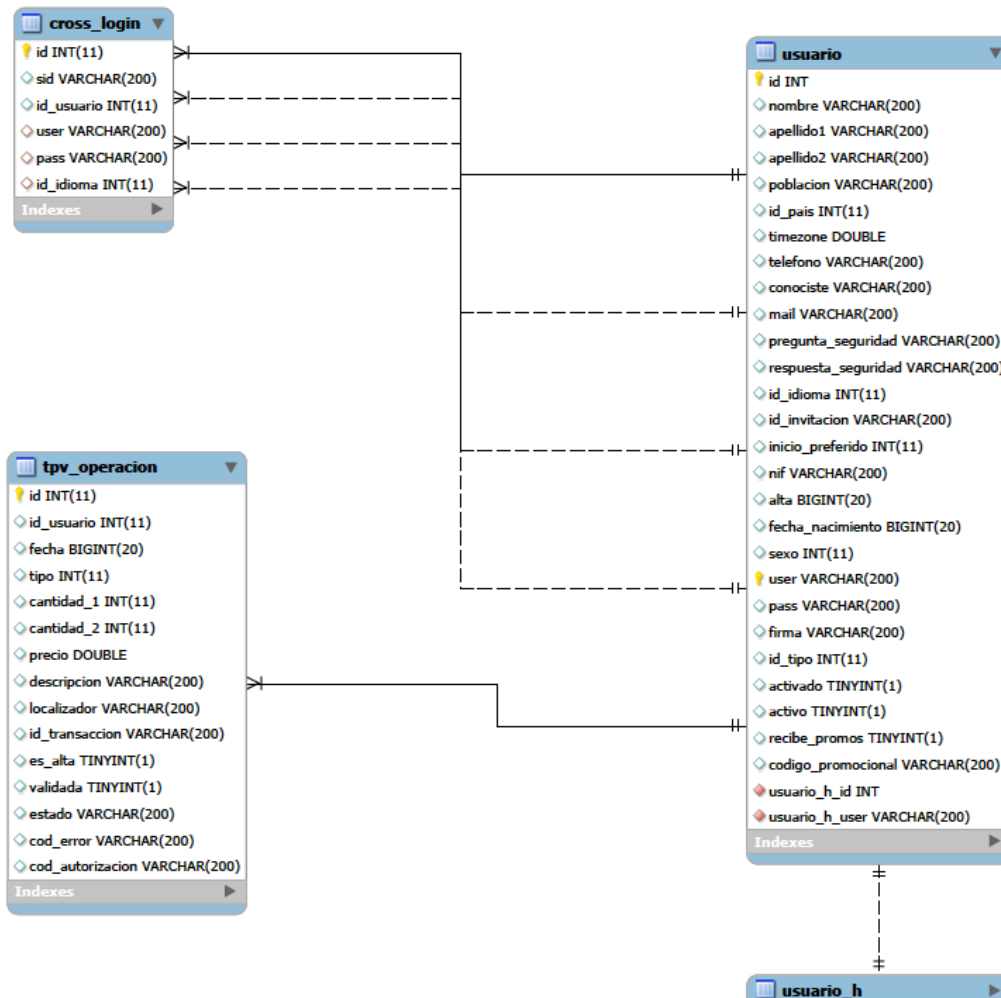


Donde el bloque de la petición AJAX se estaría repitiendo hasta que el usuario cambiase la página o cerrase el navegador.

## 3.2 Modelo de datos

A continuación se muestran diagramas E-R que definen partes del modelo de datos que emplea la aplicación.

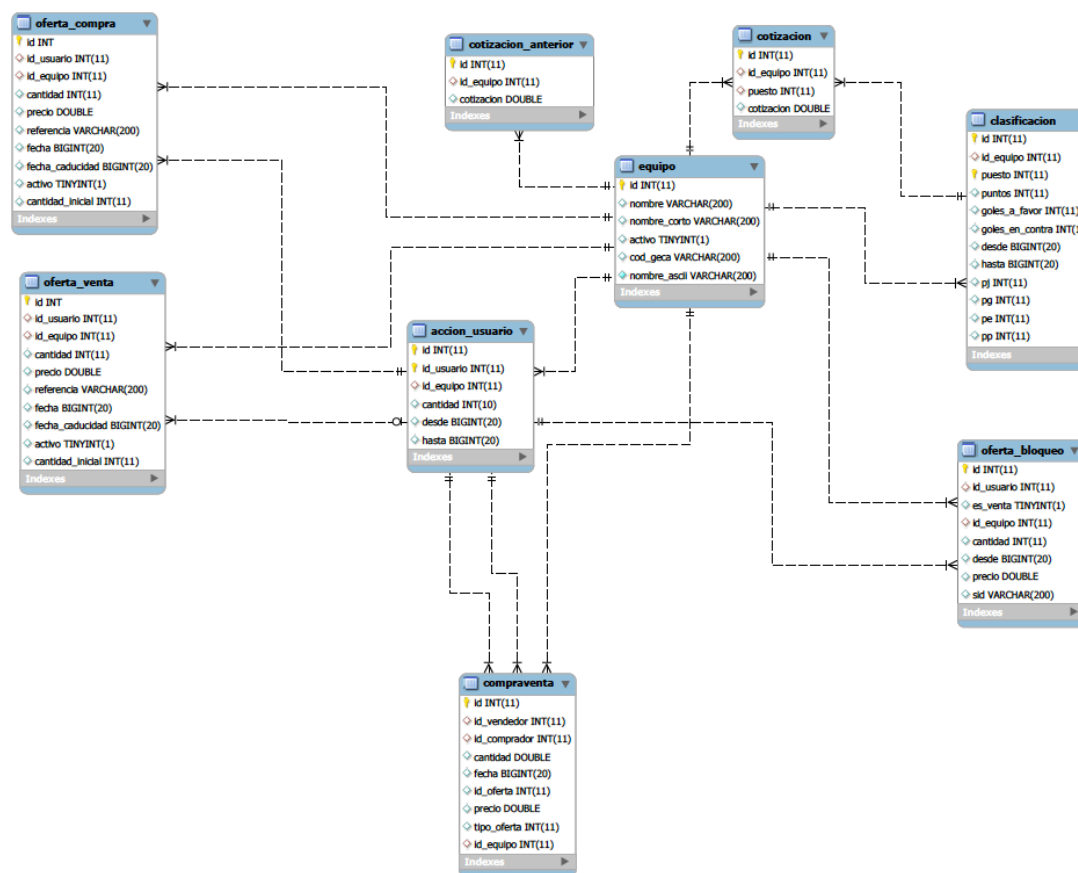
### 3.2.1. Usuarios



La tabla usuario contiene, obviamente, la información sobre los usuarios registrados en el sistema. Las operaciones que los usuarios realizan con el banco a través de la pasarela de pago quedan registradas en la tabla tpv\_operacion, independientemente de si son aceptadas o rechazadas por el banco. Esta tabla se utiliza para añadir las bolsas correspondientes cuando la respuesta del banco es positiva.

Por último, la tabla cross\_login sirve para realizar el cambio de sesión de usuario entre las diferentes ligas del sistema, ya que Tomcat no permite compartir sesiones entre contextos.

### 3.2.2. Acciones



Se podría decir que la tabla `accion_usuario` es el centro del sistema. Cabe destacar que cada registro de esta tabla no representa una acción como cabría esperar, sino la cantidad de acciones que un usuario concreto tiene de un equipo concreto.

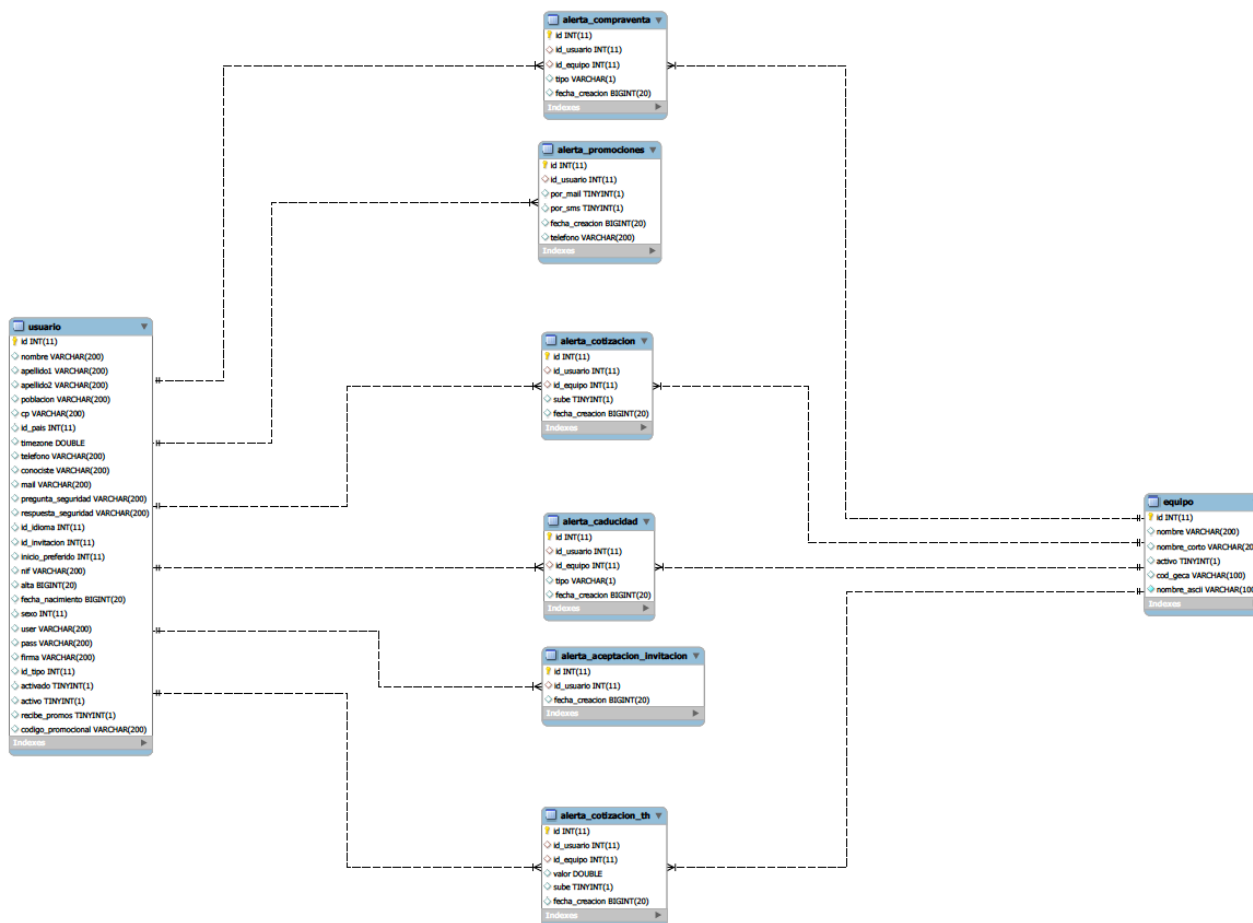
Los usuarios pueden publicar ofertas de compra y de venta, que quedarían registradas en las tablas `oferta_compra` y `oferta_venta` respectivamente y realizar transacciones de compraventa con sus acciones. Cada operación de compraventa genera un registro en la tabla correspondiente (`compraventa`) por cada usuario involucrado en la misma.

Al mostrar información sobre su bolsa al usuario, esta siempre se organiza en función de la clasificación actual de los equipos.

Finalmente, las acciones de cada equipo tienen un valor de cotización que indica el valor que otorgan los usuarios a esas acciones y sirve como referencia a la hora de operar. El valor de la cotización anterior se emplea como referencia para el envío de alarmas por correo electrónico.

Todas las tablas relacionadas con la compraventa tienen un equivalente donde se almacena un histórico de todos los movimientos y ofertas publicadas.

### 3.2.3. Alarmas



Las tablas relacionadas con las alertas son bastante simples, cada tipo de alerta tiene su tabla. Se considera que una alerta de “sube la cotización” es lo mismo que una alerta de “baja la cotización”.

Cada usuario puede configurar las diferentes alertas para los distintos equipos introduciendo los valores que crea oportunos. Exceptuando las ofertas de compraventa, el resto de alertas se eliminan una vez han saltado.

Como en los diagramas anteriores las tablas de alarmas tienen sus correspondientes tablas históricas donde se almacenan todas las alertas creadas por los usuarios junto con su fecha de envío.

## 4. Implementación

En este apartado se va a definir el proceso de implementación que se ha empleado además de sugerir mejoras al mismo. Se hará mención especial como un punto aparte el método de acceso a los datos por considerarlo una parte interesante del proceso de diseño e implementación.

### 4.1. Interacción con la base de datos

El acceso y manipulación de los datos de la aplicación se realiza a través de objetos DAO (Data Access Object), encargados de las inserciones y actualizaciones y DMO (Data Management Object) que permiten manipular los datos creando objetos para representarlos. Además se han generado unos objetos mixtos llamados “Listados” que se encargan de obtener grandes cantidades (listados) de DMOs que cumplan ciertos requisitos predefinidos y que, además de obtener los datos y construir los DMOs correspondientes, permiten realizar el borrado de registros.

Para que el acceso a la base de datos sea lo más transparente al desarrollador, se ha creado un software auxiliar que construye el código fuente en java de todas las clases necesarias, además de las sentencias de creación de tablas en MySQL.

#### 4.1.1. Plantillas para la generación de código

Las plantillas para la creación de código son archivos XML sencillos que deben definir el objeto java y mapear sus atributos en los de una tabla relacional. Además, al final, si se considera necesario, se pueden definir unas funciones de acceso a los datos que generaran las consultas más habituales.

A continuación, fragmentos de una plantilla de generación de código:

```
<bdmp-structure>

<class-name>Usuario</class-name>

<class-package>#####</class-package>

<table-name>usuario</table-name>

<prim-key-field name="id" java-type="int" db-type="int" db-field="id"/>

<field name="nombre" java-type="java.lang.String" db-type="varchar(200)" db-field="nombre"/>

<field name="apellido1" java-type="java.lang.String" db-type="varchar(200)" db-field="apellido1"/>

<field name="apellido2" java-type="java.lang.String" db-type="varchar(200)" db-field="apellido2"/>

...

<to-string>apellido1+ " "+apellido2+" ", "+nombre</to-string>

<listing>

<related-field name="id" type="&gt;"/>
```

```
<related-field name="activo" type="="/>
```

```
</listing>
```

...

```
<listing>
```

```
<related-field name="activado" type="="/>
```

```
<related-field name="alta" type="&lt;"/>
```

```
</listing>
```

```
</bdmp-structure>
```

Como se ve, se define un nombre para la clase, su ubicación en el classpath y el nombre de la tabla relacional correspondiente. A continuación se definen los diferentes atributos. Para terminar se incluyen los métodos de listado, cada nodo listing define un método.

#### 4.1.2. Código generado automáticamente

A partir de cada una de estas plantillas se generan varias clases, las más importantes son, como se ha comentado, los DAO, DMO y listados. Un ejemplo de uso:

```
...
UsuarioListados listado = new UsuarioListados(con);

UsuarioDAO udao = new UsuarioDAO(con);

UsuarioDMO[] lista = new listado.getByActivadoFromAlta(true, alta);

for(UsuarioDMO u: lista){
    System.out.println(u.getNombre);
    u.setNombre("UPNA");
    udao.update(u);
}
...
```

El fragmento de código anterior, obtiene una lista de usuarios que están activados a partir de una fecha de alta determinada, imprime su nombre de pila por la salida estándar y lo sustituye por “UPNA”. Nótese que la instrucción “u.setNombre()” no significa la actualización de los datos hasta que no se ejecuta el método “update()” del DAO.

## 4.2. Software propietario

Dado que el desarrollo para completar la funcionalidad del sitio web es muy extenso, no es el objetivo de la memoria revisar la totalidad de las líneas de código desarrolladas. La memoria se centra en el funcionamiento general del mismo y en aquellos componentes que pueden suponer, a priori, un cuello de botella en la ejecución de las peticiones y, por ende, ralentizar el portal o demandar una necesidad de recursos físicos desmedido, además de otros puntos de interés.

#### 4.2.1. Servlets de peticiones de información

Este apartado se centra en la petición de las últimas cotizaciones de las acciones de cada equipo del juego. Dicho servlet, actualmente obtiene la información de la base de datos, y la cachea en memoria usando un mecanismo de resolución de nombres JNDI.

El uso de la resolución de nombres está pensado en Java para acceder a recursos del sistema cuyo origen e implementación se desconocen. La propia resolución de nombre, puede llevar varios ciclos de proceso de la máquina por petición por lo que habría formas estándar de implementar la cache de una forma más ligera.

Una de ellas sería hacerlo con tablas en memoria del propio gestor MySQL. Dado que el gestor de la base de datos retiene en su memoria las últimas consultas realizadas, el acceso continuado a estas tablas sería muy rápido y de forma natural y común a las otras consultas realizadas.

Mediante el uso de procedimientos almacenados en la propia base de datos, se podría mantener actualizada esta tabla cada vez que se inserta algo en las tablas de operaciones cruzadas manteniendo viva la cache y quitando la sobrecarga que implica su recálculo.

#### 4.2.2. Utilidad de compraventa

Es la utilidad que se emplea para ejecutar una transacción de compra de acciones. Puntos críticos encontrados en su implementación:

- La operación de compra debe descontar las acciones reservadas por el usuario y que no hayan sido objeto de compra. Existe una posible carrera entre hilos de ejecución dentro del programa que, en algunas circunstancias, podrían dejar datos inconsistentes. Por tanto, se hace necesario eliminar esa posibilidad, la cual, una vez más, tiene alta probabilidad de ocurrir con el elevado número de usuarios jugando concurrentemente.

Dada la escalabilidad deseada en el programa, la forma de hacerlo es bloqueando el registro correspondiente en la base de datos. Esto se hace de forma simple, concretamente en MySQL usando la sentencia `SELECT FOR UPDATE` del registro a modificar. De esta manera, el bloqueo se reduce al momento en que la compra concurrente es del mismo equipo y, exclusivamente, al momento de modificar el registro de la tabla correspondiente.

- La implementación de la operación de compra lleva implícito la reserva de unas acciones al mejor precio durante un cierto intervalo de tiempo. Pasado ese intervalo de tiempo sin ejecutar la operación de compra, la reserva caduca. Es importante reducir el tiempo de ejecución del servlet al mínimo de manera que, dicho control, lo haga un “demonio” independiente que se ejecute regularmente y de forma desligada del servlet. Esto se puede hacer implementando un “singleton” estático que se lanzaría durante la inicialización del servlet.
- El control del tiempo de reserva se realiza en el navegador mediante la ejecución de código JavaScript. Es importante tener en cuenta que, si el navegador tiene deshabilitada la opción de ejecución de JavaScript, no se le debe dejar “comprar” o bien el servlet debe tener en cuenta que pueden llegarle peticiones caducadas (al no poderse ejecutar el JavaScript, no habrá caducado la reserva en el navegador,

por lo que el usuario puede finalizar la transacción con el tiempo excedido) lanzando un error que indique que la compra no se ha realizado.

- Optimización del código Java encargado de generar las queries SQL. Así, se mejoraría el rendimiento de las transacciones Java y minimizaría el uso de la memoria. Son pequeñas cosas que en una ejecución de centenares de miles de veces en lo que esté activo el servidor pueden tener una incidencia apreciable.

La recomendación más usual es poner las queries en variables estáticas, de forma que no se tenga que completar cada vez que se ejecute.

#### **4.2.3. Refresco de cotizaciones**

Esto se hace por medio de JavaScript AJAX, el cual, periódicamente solicita a un servlet que le envíe el valor de las cotizaciones de todos los equipos, el cual es calculado a partir del valor de las últimas operaciones cruzadas.

En la actualidad, el periodo de refresco ha sido fijado en 5 segundos y, según los cálculos realizados, el documento XML que se envía con la información de cotización es aproximadamente de 1,7 Kbytes.

Este refresco, en un punto importante a tener en cuenta a la hora de dimensionar las necesidades de ancho de banda dado que, un usuario registrado que acceda al juego, simplemente por el hecho de estar conectado, demandará una atención periódica importante cargando tanto la red como los servidores que deberán procesar y preparar la información presentada.

Sería recomendable ajustar el periodo de refresco a un valor más elevado de forma que el sistema pueda soportar un mayor número de usuarios concurrentes sin llegar a comprometer la usabilidad del juego.

#### **4.2.4. Sistema de Alertas**

El sistema de alertas trata de dar cobertura a las especificaciones del usuario que requiere que se le notifique vía correo electrónico la ocurrencia de ciertos eventos. Los principales son cuando se cruza alguna de las órdenes de venta que tenía en el sistema o cuando las cotizaciones de ciertos equipos marcados por él han variado de valor.

Esto se hace con programas específicos lanzados mediante un “cron” del sistema operativo. El “cron” es una facilidad del sistema operativo que permite programar tareas específicas en momentos determinados del día. Las reglas pueden ser muy complicadas, pero sólo basadas en términos horarios. Una simple sería cada 2 minutos, una complicada podría ser los minutos pares de los horas impares entre las 5 y 9 de la noche de los días laborables.

#### **4.2.5. Servlet de administración**

Además de los servlets con la lógica del juego, el sitio web cuenta con un área de acceso restringido a la que denomina “herramienta de administración”.

Aquí se han colgado algunas herramientas que sirven para controlar la bondad de la ejecución del juego como puede ser que no haya descuadres en las operaciones cruzadas



y páginas donde se pueden consultar las órdenes puestas en el sistema ya sea por horarios, por un usuario concreto...

Adicionalmente, se han añadido componentes para sacar estadísticas de las páginas visitadas en el servidor y está previsto añadirse funcionalidad para monitorizar el estado de salud del servidor (uso de memoria, ocupación de disco, carga de sistema...).

Las herramientas de monitorización del sistema las incorporará el proveedor de hosting y, en su defecto, se pueden incorporar componentes ya hechos y estándares que no impliquen desarrollos como puede ser Nagios (por mencionar uno de software libre).

En cualquier caso, seguramente será oportuno crear ciertos servlets de control que permitan a los sistemas alertar a los operadores que algunos de los componentes software no está funcionando correctamente.

### **4.3. Proceso de desarrollo**

Durante el desarrollo del proyecto no se ha empleado ninguna metodología formal ya que el equipo es pequeño y trabaja de manera conjunta. No obstante, este proceso se podría englobar dentro del grupo de metodologías ágiles.

A continuación se explica la forma de trabajar que se ha empleado a lo largo del proyecto.

#### **4.3.1. Proceso actual de desarrollo**

Se podría decir que el proyecto carece de metodología formal en cuanto a desarrollo se refiere:

- El equipo cuenta con una persona que se dedica a maquetar las páginas usando Adobe Photoshop. El resultado es traducido a HTML por otro especialista y, finalmente, el equipo de programación desarrolla la capa dinámica incorporando al HTML los tags específicos de velocity.
- Por otra parte, tras las pruebas realizadas surgen cambios, corrección de errores... El equipo en conjunto realiza reuniones con frecuencia, al menos, semanal para informar al resto de su evolución.

Es muy importante establecer un sistema de control de versiones de software que permita el trabajo concurrente en el mismo componente o fichero. Para esto se usa “Subversion” conjuntamente con “Tortoise” (su interfaz gráfica) que se integra en la mayoría de las herramientas de desarrollo Java así como en el explorador de fichero de Windows.

Este sistema permite a dos usuarios acceder y modificar el mismo fichero siendo capaz de incorporar, de forma automática, las revisiones de cada uno de ellos, o bien, notificar de las colisiones que él no pueda resolver y no quede más remedio que resolverlas por el programador, evitando así que un programador pise a otro sin tener constancia de ello.

- Por otro lado, se hace necesario disponer de un sistema de ticketing de las incidencias del software para poder automatizar su registro y evolución de una forma centralizada. Para ello se propone “bugzilla” que es una de las herramientas de uso más extendido en la comunidad de programadores Java a nivel mundial.

#### 4.3.2. Mejoras al proceso de desarrollo

Atendiendo a lo anterior y, debido a que el juego requiere una evolución continua (algo que no mejora pronto deja de ser bueno), en algunos casos puede suponer cambios estructurales importantes en sus componentes básicos, por tanto, se hace necesario ir incorporando metodologías de desarrollo adecuadas a la envergadura del proyecto.

Puntos a tener en cuenta son:

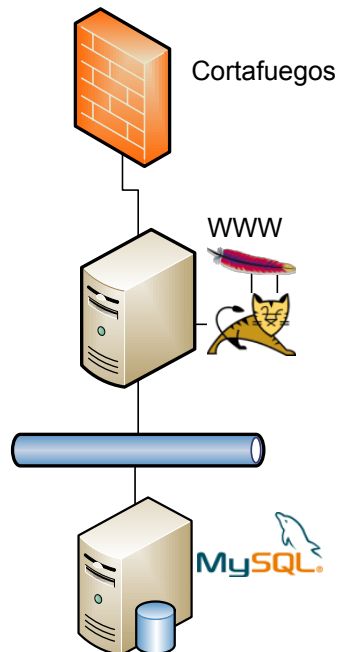
- Proporcionar a cada desarrollador un entorno de pruebas unitarias individualizado. Esto supone que cada uno de ellos podría, en caso extremo, tener una versión completa del juego donde iría probando los cambios que vaya realizando.
- Disponer de un entorno de integración y calidad. Estos consiste en una réplica a escala de la misma arquitectura de producción donde se puedan ir probando los componentes desarrollados por todo el equipo de forma conjunta así como realizar pruebas funcionales controladas (como son la recuperación ante caídas, fallos de red...).
- Incorporar el sistema de control de versiones de software y gestión de incidencias en la operativa diaria tanto de los desarrolladores como de las personas dedicadas a pruebas.
- Disponer de un entorno de compilación centralizado donde se crearían las nuevas versiones de software para ser instalado tanto en el entorno de pruebas como, en el futuro, en los sistemas de producción. Al hacerlo en un entorno específico y diferente de cualquier desarrollador se evitan, en tiempo de compilación, errores derivados de configuraciones y librerías específicas que no aparecerían hasta el momento del despliegue en el entorno de producción.
- Incorporación de reglas de estilo en la programación de manera que el código producido sea fácilmente legible por cualquier programador del equipo.

## 5. Implantación

### 5.1. Estructura del sistema

En esta sección se explicarán las funciones de los diferentes componentes del sistema partiendo de un esquema básico para terminar en la estructura elegida para la implantación del sistema.

Un sistema básico que se podría emplear para la puesta en marcha del sistema sería el siguiente:



Un servidor web encargado de servir las páginas y realizar el cálculo para mostrar los datos correctos y otro servidor de base de datos (estos servidores podrían convivir en una sola máquina pero, por clarificar el ejemplo, se muestran por separado).

La implementación física de los componentes descritos anteriormente se hace mediante la combinación de tres elementos de software, que se ejecutan en tres niveles o escalones sucesivos:

- **Servidor Apache:** Este componente es el encargado de procesar las peticiones de las vistas y componentes AJAX ejecutados en los navegadores. Dado que la mayor parte de los componentes son dinámicos, su principal misión es redirigir las peticiones a los servlets que se ejecutan en el escalón siguiente.
- **Servidor Tomcat:** Este componente es el que aloja los diferentes servlets y genera las vistas “velocity” que retornan a los navegadores solicitantes a través del Apache. Por la naturaleza del juego, es necesario que el usuario mantenga una sesión activa mientras esté conectado al mismo. Tomcat también es el gestor de los datos de sesión que tienen que ser compartidos por las distintas vistas/páginas que componen el sitio web.
- **Servidor MySQL:** Usado como la capa de persistencia de los datos necesarios para el juego: datos del usuario, preferencias del mismo, órdenes de compra/venta...

### 5.1.1. Servidor Apache

A diferencia de otros sitios web donde Apache es usado para proceso dinámico de contenidos, ya sea vía CGIs o PHP, en este caso se reduce a dos cometidos exclusivos:

- Proporcionar los contenidos estáticos de las vistas. Aunque se ha dicho que las vistas se generan en su totalidad de forma dinámica, hay ciertos componentes realmente estáticos. Se han identificado exclusivamente dos:
  1. Hojas de estilo o páginas CSS. Estas son usadas por el navegador para fijar los atributos de presentación a los distintos componentes de las vistas HTML: fuentes de caracteres, tamaños, colores, fondos...
  2. Objetos que forman parte de las vistas HTML y son independientes del idioma elegido como son las imágenes, logotipos, iconos...
- Interfaz de comunicaciones para el protocolo cifrado SSL o, lo que es lo mismo, protocolo HTTPS. Este protocolo debe ser implementado para la participación segura en la parte registrada del juego. Dado que Tomcat no proporciona un mecanismo estándar para las comunicaciones cifradas, lo habitual es que Apache realice esta tarea. Evidentemente, una vez usado como frontal de este protocolo, también es usado para gestionar las comunicaciones no cifradas del protocolo HTTP.

El servidor Apache está instalado en un servidor con sistema operativo Linux RedHat Enterprise Server 5. Se ha creído conveniente revisar algunos parámetros de la configuración estándar antes de poner el sistema en producción; a destacar:

- Modelo de gestión de componentes. Esto lo determina la combinación hardware/software usado. En el caso estudiado debe ser “modo worker” para minimizar el impacto del uso de memoria necesitado y sacar provecho de la totalidad de procesadores disponibles en el servidor.
- Modulo de reescritura de URLs, “mod\_rewrite”. Este módulo, que ha de ser activado de manera explícita, se encarga de traducir la URL solicitada a la ubicación real del recurso para poder ofrecer a los usuarios y a los robots de búsqueda URLs más sencillas. Para su correcto funcionamiento es necesario incluir una entrada por redirección. Cada entrada incluye una expresión regular y la URL real que se servirá si la solicitud verifica la expresión regular correspondiente. A no ser que se indique lo contrario estas traducciones son transparentes al usuario, es decir, no existe redirección aunque en realidad se esté sirviendo el recurso desde una URL diferente a la solicitada.
- Número máximo de conexiones concurrentes. Limita, por configuración, el número de peticiones que pueden ser atendidas de forma concurrente por el servidor. Tenía el valor estándar de instalación, que es de 150. Para elevar este valor, hay que estimar el consumo de memoria RAM necesitado para las peticiones y los límites físicos de sockets de comunicaciones que un proceso puede abrir. Para elevar este último puede que sea necesario modificar los valores del sistema operativo (por defecto establecido en 1024).
- Tiempo de espera entre peticiones. Establece el tiempo de espera máximo que el servidor esperará antes de cerrar el canal con un cliente, antes de atender a un cliente distinto. El valor depende del uso que se vaya a hacer el servidor, dado que a veces, el mismo cliente hace una segunda petición pocos segundos después de haber realizado la primera. Por nuestro análisis, dado el alto número de

transacciones concurrentes que debe soportar el servidor el valor debería ser 0, de forma que tras atender una petición, el agente quede libre para atender otra de inmediato, puesto que con muy alta probabilidad será de un cliente distinto.

- Funcionalidad KeepAlive. Por la misma razón anterior, esta funcionalidad debería estar desactivada.
- Activar el módulo de caché de forma que los contenidos realmente estáticos sean resueltos desde la memoria en lugar de volver a leerlos en disco.
- Configurar los directorios que alojan los sitios estáticos según las recomendaciones de Apache de forma que se eliminen los tiempos muertos de resolución de enlaces simbólicos y resolución de permisos de accesos.

### 5.1.2. Servidor Tomcat

El servidor Tomcat es la extensión de Apache usada para la ejecución de los componentes del estándar Java Servlet 2.5. Es el componente mínimo requerido para la ejecución de la lógica de negocio del juego puesto que, justamente, ésta se ha implementado siguiendo el estándar Java Servlets.

Las peticiones le llegan a los servlets provenientes del servidor Apache. La conexión entre ambos se hace a través del módulo proxy AJP, que es el mecanismo propuesto por Apache para interconectarlos.

Tomcat incorpora otras ventajas fundamentales para el programador como es un sistema de conexiones con la base de datos, mecanismo de resolución de nombres (JNDI)....

Adicionalmente, se han encontrado algunas debilidades en la configuración por defecto de Tomcat que deberían tenerse en cuenta, todas ellas relativas a la configuración del pool de conexiones con la base de datos:

- El fichero de configuración contiene los passwords de acceso de la base de datos en texto plano. Es así dado que se emplea el mecanismo estándar y este no permite el paso de argumentos cifrados. Sería imprescindible, al menos, restringir los permisos de accesos al fichero de configuración de Tomcat de forma que limitase el acceso no controlado a dichas claves.
- El sistema operativo va a limitar el número de sockets que el proceso puede abrir a 1024. Por otro lado, este valor debería ser sensiblemente inferior al valor de conexiones de MySQL por la propia naturaleza del pool y no debería superar el número de conexiones que Apache puede establecer con este servidor.

### 5.1.3. Base de datos MySQL

Se propone como gestor de base de datos MySQL Community Edition, que es una versión de software libre. Las características básicas de este gestor son que, además de implementar el lenguaje SQL, permite elegir entre varios motores de bases de datos, cada uno con sus peculiaridades técnicas en consideraciones de fiabilidad y rendimiento.

En el mismo gestor se crearán, al menos, dos bases de datos distintas: una con el contenido de los datos de los usuarios y otras con la información propia del juego. Esto se ha hecho así para que, si en el futuro se requiriese que el juego sea “multi liga”, tener una base de datos distinta para cada juego.

Los detalles técnicos más relevantes a tener en cuenta en cuanto a rendimiento de la base de datos son:

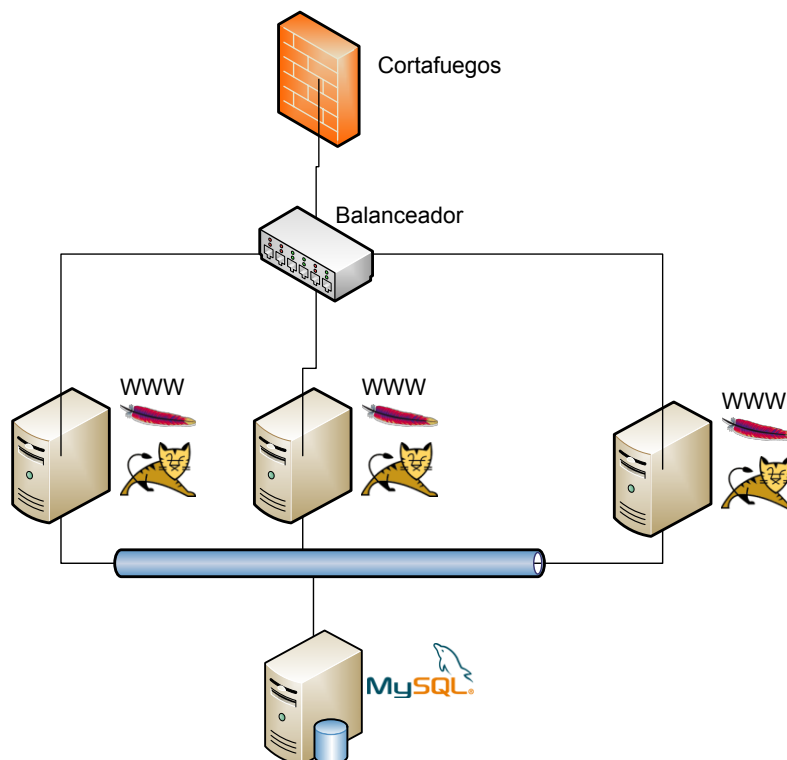
- Todas las tablas de la base de datos se han creado usando el motor “InnoDB”. Éste es el motor transaccional que permite, a su vez, establecer relación de integridad en el modelo de datos.
- El límite físico de 1024 conexiones impuesto por el sistema operativo. Téngase en cuenta que, en este caso, este límite incluye los archivos físicos de comprenden cada una de las bases de datos por lo que el número de conexiones reales disponibles será sensiblemente inferior, especialmente si el gestor de MySQL sirve varias bases de datos y estas contienen muchas tablas e índices (como es el caso).

Teniendo en cuenta los requisitos de escalabilidad y alta disponibilidad del sistema, la estructura explicada anteriormente se queda obviamente corta.

#### 5.1.4. Escalabilidad

La plataforma no sólo debe ser capaz de atender inicialmente un elevado número de usuarios concurrentes sino que, además, debe ser capaz de ir creciendo en la medida que se incrementa el registro de usuarios, sin necesidad de tener que alterar su desarrollo. Esta característica se conoce como escalabilidad.

La forma más usual de escalar el sistema es replicar el número de servidores que atienden las peticiones. Una estructura básica es la mostrada en la figura siguiente.



Junto a las réplicas de los servidores Apache+Tomcat, aparece un elemento nuevo, conocido como balanceador. El cometido de este componente es repartir las peticiones que llegan de los usuarios entre los servidores disponibles.

Al conjunto de servidores replicados y conectados de esta manera se conoce cómo granja de servidores y en principio pueden crecer en número según se vaya necesitando.

El trabajo del balanceador es transparente al usuario y, en caso de fallo de uno de los servidores, las peticiones son redirigidas al resto. La configuración más usual del balanceador es que mande las peticiones al servidor más desahogado, es decir, el que menos tiempo tarde en responder, de todos los disponibles.

Hay que tener en cuenta un punto muy importante: el juego está basado en servlets y para poder jugar hay que mantener la sesión del usuario en todo momento. Al poner el balanceador por delante de Tomcat, puede ocurrir que una petición llegue a un servidor y, la siguiente petición del mismo usuario, llegue a otro distinto.

Hay dos formas de resolver este problema. La primera y más fácil, es configurar el balanceador para que todas las peticiones de un mismo usuario vayan al mismo Tomcat. La segunda, algo más compleja de probar pero con ventajas evidentes, es configurar los Tomcat en clúster de forma que el objeto de sesión sea compartido por todos ellos.

Esto aseguraría que, ante un eventual fallo de un servidor Tomcat, el usuario registrado no se percataría del problema mientras que, el método simple, ante la caída de uno de los servidores, todas las sesiones alojadas en él deberían volver a loguearse en el sistema e iniciar una nueva sesión en otro de los servidores activos.

Si bien el sistema descrito aquí asegura una cierta escalabilidad sigue presentando un punto singular de fallo que no es otro que, si se cae el servidor MySQL, el sistema dejaría de funcionar hasta que no se levantara el mismo. Esto da pie al otro punto a tener en cuenta: la alta disponibilidad.

### **5.1.5 Alta disponibilidad**

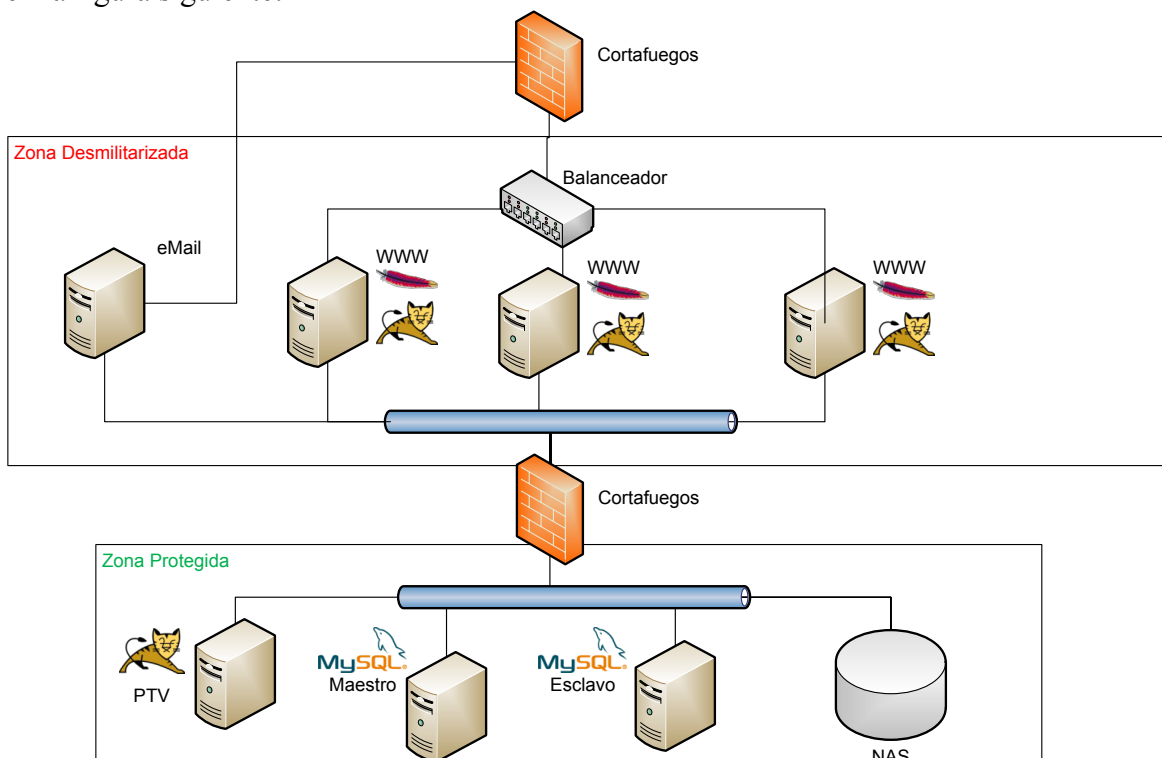
Se entiende por alta disponibilidad a un sistema que es capaz de seguir trabajando la mayor parte del tiempo aunque alguno de sus componentes falle. La mayor parte del tiempo, ante el fallo de un elemento, lo que se suele hacer es que otro asuma automáticamente el trabajo del que ha dejado de funcionar y, desafortunadamente, eso no ocurre de forma automática.

Si el tiempo de reacción fuese exactamente cero, se hablaría de un sistema tolerable a fallos pero no hay muchos en el mercado y los que hay son muy especializados y extraordinariamente caros.

Lo que se suele intentar es limitar o reducir el tiempo de caída al valor más pequeño de parada que nuestro sistema pueda soportar. Se suelen medir de muchas maneras aunque el método más usual es el grado de fiabilidad: 99,9%, 99.99%, 99.999% A más alto grado de fiabilidad, menor es el tiempo de recuperación del sistema pero, también, más caro, al necesitar un mayor número de elementos redundantes.



El sistema de alta disponibilidad propuesto para el entorno de producción es el reflejado en la figura siguiente:



En el esquema se ha añadido ya alguno de los elementos de seguridad a tener en cuenta para asegurar que los datos de los usuarios están protegidos de ataques externos y que serán fundamentales para pasar las auditorías impuestas por las compañías de medios de pago por internet y la LOPD entre otras.

Las dos zonas, conocidas como zona desmilitarizada y zona segura, están separadas por un cortafuegos. Esto es una medida de seguridad extra que aseguraría el sistema ante un hipotético ataque que lograra entrar en uno de los ordenadores de la zona desmilitarizada. Por otra parte, no impide el acceso directo a internet desde la zona segura.

Se aprecia en el diagrama que se han separado la granja de servidores Apache y Tomcat de los servidores que acogen la base de datos. Las consideraciones hechas en la sección anterior para la escalabilidad de Tomcat siguen siendo válidas y deben ser tenidas en cuenta en esta nueva infraestructura.

Las novedades de la infraestructura propuesta son:

- Se pone un servidor Tomcat en la zona segura. Su cometido será responder a las transacciones de los medios de pago de forma que, el propio servlet, estará en ejecución dentro del entorno seguro. De esta manera, los datos de las tarjetas de pago, cuentas corrientes... nunca estarán, ni siquiera en la memoria, de los ordenadores de la zona desmilitarizada. Lógicamente, la base de datos que contenga esta información tampoco estará accesible desde dicha zona.
- Se dispone de un clúster para ejecutar MySQL server. En principio se han analizado las alternativas existentes para montar este clúster y la más fiable, sin



necesidad de cambiar los programas realizados hasta la fecha, es usar un clúster gestionado por el propio sistema operativo Linux. De esta manera, se tendrían dos instancias de MySQL server en ejecución, una maestra y la otra esclava. Si la maestra cae, el clúster activa el servidor de esclavo de forma automática. El tiempo comprometido de recuperación del sistema debe ser inferior a los 3 minutos aunque en la práctica puede ser inferior al minuto.

Para lograr mantener actualizados el maestro y esclavo en tiempo real se requiere añadir una cabina de almacenamiento de discos. Este sistema de almacenamiento es compartido por todos los elementos de la red, existiendo varias variantes del mismo. En nuestro caso, para la base de datos se empleará una configuración “NAS” usando para la conexión fibra óptica, dado que el tiempo de acceso es similar al que se obtendría con discos montados localmente en el servidor.

- Debido al sistema de alerta, se presume que se va a dar gran cantidad de notificaciones vía correo electrónico a los usuarios por lo que se ha añadido un servidor email Relay dedicado, cuyo cometido será el envío de los email a los producidos a los usuarios. Si el volumen de email es muy alto, se deberá tener en cuenta el posible impacto en el ancho de banda.

Se propone un único servidor de correo, pero, si el envío de email se quiere tener también en alta disponibilidad, se debería incluir un clúster para darle alta disponibilidad a este sistema. No se ha incluido ya que el tiempo de failover en este sistema no es crítico, pudiéndose gestionar manualmente a través de la vitalización de la máquina donde opera.

La interconectividad de los servidores entre sí será, al menos, de GigaBit Ethernet, para asegurarnos tiempos de latencia mínimo entre ellos.

La configuración en alta disponibilidad obliga a que la electrónica de red, aunque no se haya representada el diagrama, esté redundada. Los servidores deberán tener las tarjetas de red duplicadas.

Tampoco se ha contemplado la alta disponibilidad del Tomcat dedicado a la gestión del cobro de bolsas. En el diagrama se ha dibujado un único proceso dejando, aparentemente, un punto singular de fallo. La solución es replicar este componente en las mismas CPUs de clúster donde se aloja MySQL, de esta manera, o bien integrar este componente dentro de gestión del clúster Linux o bien crear dos instancias de Tomcat (una por CPU) formando un mini-clúster propios de Tomcat. Cualquiera de estas alternativas, frente al uso de servidores dedicados, tiene la ventaja de que se ahorra el número de máquinas necesarias.

## 5.2. Parámetros de configuración

Para conseguir la configurabilidad requerida por el cliente se intenta abstraer al máximo la aplicación de manera que si hubiese que realizar algún tipo de portabilidad o adaptación a otro tipo de deporte o competición, los cambios en el código sean mínimos.

Parámetros de configuración, como datos de acceso a bases de datos, rutas hacia los archivos necesarios... además de datos relativos a la competición como número de equipos y número de jornadas entre otros se recogen en un archivo que se procesa al iniciar el sistema y quedan almacenados en memoria.

A continuación un fragmento del archivo de configuración (por razones de confidencialidad alguno de los datos ha sido eliminado)

```
<syrius-deploy>

<dbDriver>com.mysql.jdbc.Driver</dbDriver>
<dbUrl>jdbc:mysql://*****/lbl1</dbUrl>
<dbUser>usuario</dbUser>
<dbPass>*****</dbPass>
<db-datasource>jdbc/lbl1DB</db-datasource>
<db-datasource-2>jdbc/lbl2DB</db-datasource-2>

<action-prefix>com.berrazar.lbl.action.</action-prefix>
<adminaction-prefix>com.berrazar.adminaction.</adminaction-prefix>
<default-encoding>UTF-8</default-encoding>
<smtp-host>192.168.100.8</smtp-host>

<webapp-context>liga-bbva</webapp-context>

<!-- velocity -->
<velocity-templates-directory>/path/templates/</velocity-templates-
directory>
<idioma-textos-home>/path/textos/</idioma-textos-home>
...
</syrius-deploy>
```

## 6. Pruebas

### 6.1. Pruebas de carga

Es indudable que, los límites reales del sistema, no se pueden determinar hasta que se realicen pruebas veraces de carga. Las condiciones para la realización de éstas son:

- Realizarlas con toda la infraestructura completa, es decir, poniendo los servidores redundantes y el sistema de alta disponibilidad.
- Realizarlas usando servidores externos que prueben todos los componentes de red: cortafuegos, balanceadoras...
- Si es posible, atacar al sistema desde cuantos más frentes mejor. Las herramientas de pruebas funcionan simulando cada una de ellas un número elevado de usuario. Esto da una idea aproximada del comportamiento del sitio web pero hay que tener en cuenta que la propia herramienta de simulación está limitada por el ordenador donde se ejecuta. Si se desea simular 1000 usuarios, se puede usar un único ordenador ejecutando la herramienta de pruebas, pero es mucho mejor usar 10 ordenadores simulando 100 usuarios cada uno.

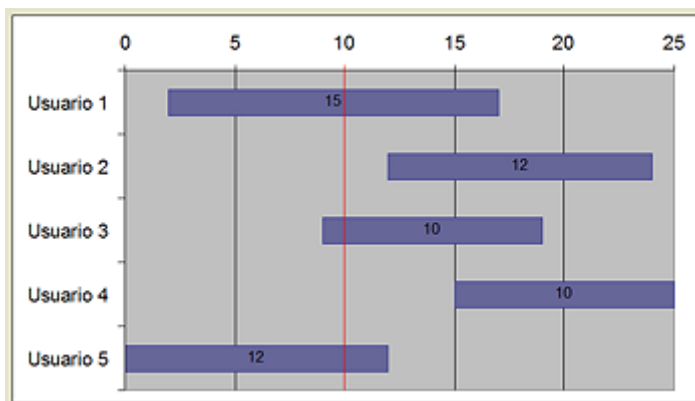
### 6.2. Análisis teórico del ancho de banda

Se realizaron dos análisis hipotéticos para intentar ver cuál es la necesidad real del ancho de banda.

El primero se basa en estimar el número de usuarios concurrentes que deberá soportar el sistema y, el segundo, cómo las características técnicas del producto influyen en el consumo de dicho ancho de banda.

#### 6.2.1. Estimación del número de usuarios concurrentes

Normalmente, cuando se habla de usuarios concurrentes, se refiere a cuántos usuarios están logueados en la aplicación durante un instante dado. Es decir, puede que una aplicación tenga N usuarios registrados, pero es muy poco probable que todos ellos operen sobre la aplicación al mismo tiempo:



Navegando por la aplicación, se puede obtener el promedio que le tomaría a un usuario realizar las funciones publicadas por ésta. Es decir, se podría saber cuál es el tiempo  $L$  que un usuario está logueado de principio a fin. Así se obtiene una estimación más aproximada:

$$C = \sum (L1, L2..., Ln) / T$$

Donde:

$C$  = Número de usuarios concurrentes.

$Ln$  = Tiempo de uso de la aplicación por el usuario  $n$ .

$T$  = Intervalo de tiempo durante el cual se realiza la medición.

Usando el ejemplo de la gráfica de Gantt anterior:

$$L1 = 15, L2 = 12, L3 = 10, L4 = 10, L5 = 12$$

Y el tiempo de medición podría ser desde  $t = 0$  hasta  $t = 25$  (tiempo durante el cual la aplicación está disponible). Aplicando a la fórmula propuesta, se obtiene:

$$C = (15 + 12 + 10 + 10 + 12) / 25 = 2.36, \text{ redondeado } \approx 3$$

Por lo que en un tiempo determinado, hay 3 usuarios concurrentes para nuestra aplicación de ejemplo.

Alternativamente, se puede calcular el mismo número usando la fórmula:

$$C = n L / T$$

Donde:

$C$  = Número de usuarios concurrentes.

$n$  = Número de usuarios totales.

$L$  = Tiempo promedio de uso de la aplicación para todos los usuarios.

$T$  = Intervalo de tiempo durante el cual se realiza la medición.

Y con el mismo ejemplo:

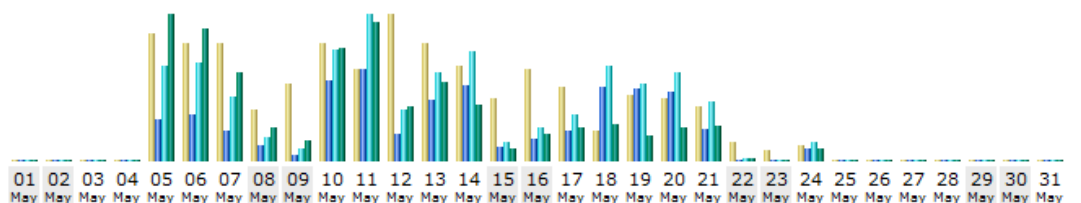
$$C = 5 \times 11.8 \text{ (el promedio de tiempo de sesión de los usuarios)} / 25 = 2.36 \approx 3$$

Se aplican estos conceptos para estimar el número de usuarios concurrentes, usando para ellos los datos estadísticos disponibles del sitio web actual del juego. Aunque se disponen de datos en mayo y junio, los cálculos emplean los datos del mes de mayo porque fue cuando se realizaron las pruebas con la liga en vigor. Se desconoce la validez del espacio muestral usado (no se sabe si los usuarios que participan en la prueba tienen el perfil del usuario al que va dirigido el juego) pero, como es el único dato de partida disponible, se dará por válido.

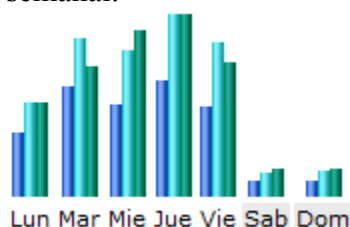
Para estos cálculos se contó con la colaboración del personal del CEIN, al que se le proporcionó acceso a la página a finales de la temporada 2009/2010, antes de su publicación definitiva.

Para el mes de mayo, se dispone de los siguientes datos:

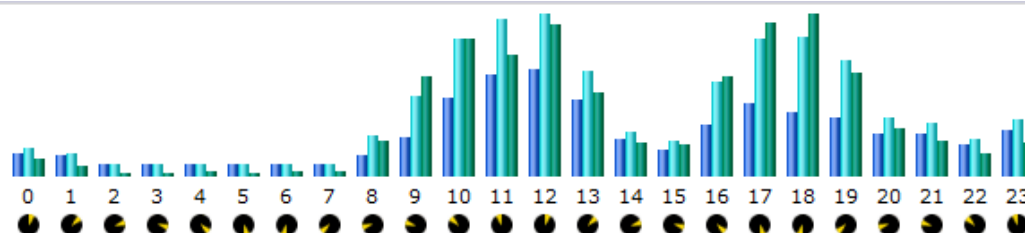
Distribución mensual:



Distribución semanal:



Distribución horaria:



Se puede observar que el patrón de comportamiento en el juego es similar a lo largo del mes, si bien a final de mes parece que hay menor actividad, posiblemente por la finalización de la liga.

Semanalmente, el grueso de las personas juega de lunes a viernes, decayendo la actividad masivamente sábados y domingos. Asumiendo, a grandes rasgos, que el 90% de los usuarios registrados usa los sistemas de lunes a viernes.

Y, finalmente, la distribución horaria presenta dos patrones similares concentrando la mayoría en dos intervalos, uno que va de 9 a 13 horas y el otro de 16 a 19, dejando a una minoría distribuirse a lo largo de la noche y madrugadas.

De las propias estadísticas de la web, para el mes de mayo se obtiene que el tiempo de conexión promedio ha sido 1384 segundos (23 minutos aproximadamente). Usando el método descrito en el ejemplo y usando la distribución semanal, se obtiene que el número de usuarios concurrentes más probables de lunes a viernes sería:

$C = 0,0024 * N$  siendo N el número de usuarios registrados.

La siguiente tabla muestra el valor estimado para el número de usuarios concurrentes en el sistema.

Usuarios Registrados	Usuarios Concurrentes
100.000	2.400
300.000	7.200
1.000.000	24.000

De las estadísticas recogidas, también se puede concluir que el tráfico más elevado se corresponde con las descargas de los valores de MiBolsa, MercadoContinuo y Cotizaciones, los tres con pesos muy similares.

Páginas-URLs (Top 10) - Lista completa - Página de entrada - Salida					
135 páginas diferentes	Accesos	Tamaño medio	Página de entrada	Salida	
/lb1/x/j.x.MiBolsa	127895	1.30 KB	20	81	
/lb1/x/j.x.MercadoContinuo	78446	2.38 KB	15	62	
/lb1/x/w.x.Cotizaciones	8304	1.30 KB	3	7	
/lb1/x/j.x.Equipo	7392	580 Bytes		3	
/lb1/x/j.x.OfertasMejores	4644	157 Bytes		2	
/lb1/x/j.x.DesbloqueaOfertas	4192	90 Bytes		4	
/lb1/x/j.x.MisMovimientos	4001	2.21 KB	1	7	
/d/lb/rup.php	3443	2 Bytes	44	44	
/lb1/stickyImg	2137	3.82 KB	4	119	
*	1503		9	9	
Otros	9710	7.98 KB	325	83	

¿Se puede asumir que la mayoría de los usuarios acceden a alguna de estas tres páginas concurrentemente? Fijándonos en el número de hits que tienen estas páginas, se puede decir que con 85% de probabilidades así sería. Por tanto, se supone que al menos un 85% de los usuarios concurrentes están transmitiendo paquetes de 1,7 KBytes (valor promedio de los 3 máximos hits).

Así pues, el ancho de banda necesario para soportar esta demanda será el valor de bits del paquete multiplicado por el número de usuarios concurrentes.

Ahora bien, los valores recogidos en la tabla anterior han sido calculados suponiendo una distribución uniforme de los usuarios a lo largo del día. Dado que las estadísticas recogidas reflejan que no es así, sino que la distribución horaria dice que el 75% de los usuarios lo hace en 9 horas, es necesario ajustar el cálculo teniendo en cuenta esto último.

Aquí entran unos cuantos conceptos de probabilidad y estadística. La distribución de Poisson es una función utilizada para modelar la probabilidad de que un evento o número de eventos ocurran a lo largo de un tiempo dado. Con una función de Poisson se puede demostrar que la probabilidad de que un número de usuarios concurrentes sea menor que  $C + (3 \times \text{SQRT}(C))$  es de 99.87%. Es decir, el pico máximo de usuarios es tal que:

$$C_{\max} \approx C + 3 \sqrt{C}, \text{ con una probabilidad de } 99.87\%$$

Así,  $C_{\max} \approx 0,0053 N$ , siendo  $N$  el número de usuarios registrados.

La nueva tabla de valores sería:

Usuarios Registrados	Usuarios Concurrentes Pico	Ancho de banda PICO
100.000	2.547	35 Mbits
300.000	16.279	90 Mbits
1.000.000	24.465	300 Mbits

### 6.2.2. Influencia del refresco en el ancho de banda

Los cálculos anteriores se basan en la estimación directa del ancho de banda del uso que hicieron los usuarios durante la prueba.

El sistema estaba configurado para enviar un refresco de las cotizaciones “en tiempo real” cada 5 segundos. Esto implica que, durante la sesión promedio del usuario de 23 minutos, se han transferido 256 veces el valor de las cotizaciones.

Teniendo en cuenta el volumen de refrescos que necesitará soportar el sistema, se ha estudiado la incidencia de esta funcionalidad en el ancho de banda demandado.

Cada vez que el usuario hace un refresco de datos, el servidor le envía un documento XML cuyo tamaño se ha medido y es de 1,7 KBytes. Suponiendo que el usuario está conectado a través de un ADSL doméstico estándar de 3 Mbits por segundo, esto supone que la información tarda en llegarle aproximadamente 5 milisegundos. Hasta los 5 segundos que vuelva a pedir el siguiente refresco, resulta que, con el ancho de banda contratado actualmente en ArSys de 9 Mbits por segundo, se podría atender sin merma de calidad a 3.000 refrescos concurrentes.

Por encima de este número, sin aumentar el ancho de banda, se producirían colisiones en la red de manera que el tiempo de transmisión de los datos hasta el usuario final aumentará. Lógicamente 5 milisegundos es un valor tan pequeño que el usuario no notaría la degradación de la red inmediatamente. Lo malo de esto es que la degradación crecerá exponencialmente con el número de colisiones de forma que, duplicar el número de refrescos sin aumentar el ancho de banda, supondría un aumento muy por encima del doble en el tiempo de transmisión.

Si se aumenta el periodo de refresco y pasa de 5 a 30 segundos, el sistema podría atender 18.000 refrescos sin merma apreciable en el tiempo de transmisión. Por lo tanto, se puede concluir que sí tiene una incidencia muy importante en el ancho de banda necesario.

A continuación, se analiza cómo podría influir ese incremento en el periodo de refresco en el ancho de banda necesario. Para la estimación del ancho de banda realizada en el apartado anterior, se utilizan los datos de uso y tráfico que el sitio web durante las pruebas, correspondiendo el máximo número de hits y tamaños de descargas a los de datos de cotizaciones. Es muy probable que esto fuese así por el bajo periodo de refresco que se estableció para las pruebas. Dado que es posible comprobar en las estadísticas que el peso de la información requerida para comprar/vender/desbloquear es sensiblemente menor que para el refresco de cotizaciones, hasta un factor 10 más pequeño.

Aunque, es difícil calcular con exactitud qué valor se hubiese obtenido si el periodo de refresco hubiese sido de 30 segundos. Pero sí se puede estimar que la necesidad de ancho de banda requerido se reduciría aproximadamente un 70%, teniendo en cuenta que no solo habría menor transmisión de paquetes de gran tamaño sino que, también, la probabilidad de que un usuario conectado estuviese en ese momento dado realizando el refresco sería más baja.

Cuanto más alto sea el periodo de refresco, más cercano estará el valor del paquete transmitido a los 100 bytes que actualmente se necesitan para las operaciones de compra.

### 6.2.3. Conclusiones

La estimación del número de usuarios medio concurrente y de pico se ha realizado con base a los datos estadísticos del sitio web recogidos durante las pruebas del mes de mayo. De estos datos, se extraen las siguientes hipótesis:

- El valor promedio de conexión del usuario con el juego es de 23 minutos.
- El 90% de los jugadores lo hace de lunes a viernes.
- El 75% de los jugadores lo hace en 9 horas del día, 4 por mañana y 5 por la tarde.

Con esto se obtiene que el valor de pico de los usuarios concurrentes usando la distribución de Poisson sería de 0,0053 por el número de usuarios registrados en el juego.

Otro resultado importante obtenido, es la gran incidencia del refresco automático a la hora del cómputo del ancho de banda necesario para atender esta demanda de usuarios. Impone una tasa de transmisión de datos que condiciona por sí sola todos los requisitos del sistema.

Dado que, hasta que no se ponga en producción, no se verá si el comportamiento real de los usuarios coincide o no con los datos extraídos durante el periodo de prueba; los parámetros de ancho de banda necesario y usuarios concurrentes deberían volver a calcularse una vez se tengan datos reales.

### 6.2.4 Cálculos reales

A continuación se van a realizar los mismos cálculos para los datos obtenidos durante el mes de septiembre de 2010. Los gráficos que aparecen a continuación tienen un formato diferente al anterior ya que las pruebas se realizaron en un entorno distinto al de producción.

Distribución mensual:



Los máximos de la imagen corresponden con los fines de semana, o los días en los que se disputaba jornada entre semana.



### Distribución semanal:



### Distribución horaria:



En las horas nocturnas disminuye la actividad, que se distribuye de manera más o menos homogénea durante el resto de horas aumentando la actividad a partir de las 7 de la tarde.

Como se puede observar estos datos difieren bastante de los obtenidos durante las pruebas. Puede deberse a que los usuarios de pruebas aprovechaban sus horas de trabajo para entrar en la página mientras que el público general aprovecha su tiempo libre para acceder a la página y jugar. No obstante, estos periodos de máxima actividad también coinciden con los momentos en los que se desarrollaba la campaña publicitaria.

Si ahora se toman los datos obtenidos y se realiza el mismo análisis:

- El valor promedio de conexión del usuario con el juego es de 15 minutos y 19 segundos.
- El 60% de los jugadores lo hace de lunes a viernes.
- El 80% de los jugadores lo hace en 14 horas del día de 7 de la mañana a 1 de la mañana del día siguiente.

Realizando los cálculos anteriores (a partir de la media de visitas por semana), se obtiene que el número de usuarios concurrentes más probables en una semana sería:

$C = 0,0015 * N$  siendo N el número de usuarios registrados.

Si se tiene en cuenta que la mayor concentración de usuarios se produce durante el fin de semana (justo al contrario que los casos de prueba) y se ajustan los parámetros en función de estos datos (tomando valores parciales de los fines de semana y ajustándose a las horas de mayor tráfico), se obtiene:

$$C=0.0078 * N$$

La siguiente tabla muestra el valor estimado para el número de usuarios concurrentes en el sistema.

Usuarios Registrados	Usuarios Concurrentes	Ancho de banda
100.000	780	10 MB/s
300.000	2.340	30 MB/s
1.000.000	7.800	100 MB/s

De nuevo, si se repiten los cálculos para usuarios y ancho de banda de pico:

$C_{\max} \approx C + 3 \sqrt{C}$ , siendo N el número de usuarios registrados.

La nueva tabla de valores sería:

Usuarios Registrados	Usuarios Concurrentes máximos	Ancho de banda máximos
100.000	864	12 MB/s
300.000	2485	36 MB/s
1.000.000	8210	120 MB/s

## 6.3 Auditorías

El sistema ha pasado tres auditorías por parte de empresas externas para garantizar la seguridad, robustez y calidad del mismo. Antes de la salida al mercado el sistema superó una auditoría de seguridad por parte de la empresa s21sec, que encontró alguna vulnerabilidad menor, una vez corregidas estas vulnerabilidades se volvió a realizar un examen que esta vez no tuvo ningún resultado negativo. Durante el primer año de vida del proyecto s21sec realizó otra auditoría de seguridad sobre el sistema en producción sin previo aviso para ver como respondía el sistema. Esta vez también fue superada sin mayores dificultades.

La empresa Medyction realizó una auditoría sobre la calidad y funcionalidad del sistema realizando alguna observación sobre posibles mejoras y cuellos de botella del sistema, estas recomendaciones fueron tenidas en cuenta dentro de la medida de lo posible.

## 7. Conclusiones y líneas futuras

A continuación, se enumeran las conclusiones de este PFC junto con unas posibles líneas futuras de trabajo.

### 7.1. Conclusiones

Este proyecto ha supuesto la implementación de un sistema de compraventa de acciones de equipos, todos los objetivos planteados en el mismo se han cumplido.

Durante el desarrollo del proyecto se ha afrontado el ciclo de vida completo del sistema, desde la captura de requisitos hasta el mantenimiento. Además se han tratado tanto aspectos de hardware, como la elección y configuración de máquinas; como software, en el proceso de desarrollo del sistema.

Para realizar este proyecto se han empleado diferentes tecnologías tanto conocidas (Java, JavaScript) como desconocidas a priori por los desarrolladores (velocity) que se han tenido que aprender para poder implementar el sistema. Además se ha desarrollado una herramienta de generación de código a partir de metadatos. Se ha trabajado en el entorno de un equipo de trabajo, empleando herramientas para facilitar la colaboración entre los miembros del mismo.

El sistema se ha implementado con éxito para la primera y segunda división de la Liga de Fútbol Profesional y cumple todos los requisitos y expectativas del cliente. Además se superan con éxito dos auditorías externas de seguridad y de calidad.

El sistema es altamente configurable y su exportación a otras ligas es relativamente sencilla.

El 25 de Agosto de 2010 el sistema fue publicado online como labolsadelaliga ([www.labolsadelaliga.com](http://www.labolsadelaliga.com)). Tras su salida al mercado fueron necesarios algunos ajustes gracias a los cuales se obtuvo un producto más estable y robusto.

A lo largo de la primera temporada del juego se generó contenido adicional para el sitio web, se introdujeron mejoras para facilitar tanto el juego como el trabajo de los administradores. Se creó una herramienta que permite a los usuarios consultar los correos enviados en nuestra web, una serie de widgets para que los afiliados pudiesen incluir en sus páginas...

Durante la primera temporada del juego se registraron más de 50.000 usuarios, se visitó la página más de un millón de veces, se vieron más de seis millones de páginas y se envió cerca de medio millón de correos electrónicos.

El sistema de compraventa de acciones consiguió una licencia como producto oficial de la Liga de Fútbol Profesional.

## 7.2. Repercusión en medios

Al tratarse de un producto novedoso su salida al mercado suscitó interés en la prensa tanto deportiva como generalista, a nivel local, diario de Navarra realizó un reportaje sobre la empresa y otros medios nacionales hicieron notas de prensa sobre el producto.



## 7.3. Líneas futuras

Una vez que el sistema ya es estable es muy fácil adaptarlo a otros deportes y ligas. Al ser un sistema de compraventa de acciones, da igual que las acciones que se venden sean de equipos de fútbol, de baloncesto o de empresas. Por lo que con un mismo núcleo del sistema se puede dar multitud de servicios que aún estarían por adaptar

Unas posibles líneas de trabajo futuras son:

- Realizar un estudio sobre la modularización del sistema y estructurar mejor los módulos existentes.
- Implementación de un juego “virtual”, que permita operar de la misma manera que el actual pero vendiendo y comprando acciones ficticias y estableciendo premios para los mejores jugadores.
- Implementación del resto de grandes ligas de fútbol.
- Implementación de ligas de otros deportes NBA, ACN, NFL...

Actualmente ya se está desarrollando alguna de estas líneas de trabajo: ya existe una versión del juego virtual y, actualmente, se está trabajando en una versión del juego para la liga mexicana de fútbol.

## 8. Bibliografía y enlaces

- INTRODUCCIÓN A LA INGENIERÍA DEL SOFTWARE. Fernando Alonso. Loïc Martínez. Fco. Javier Segovia 2005
- THE PRAGMATIC PROGRAMMER. Andrew Hunt. David Thomas ,1999
- INGENIERÍA DEL SOFTWARE. Ian Sommerville 2005.
- BUILDING WEB APPLICATIONS WITH UML. Jim Conallen 2000
- INTEGRAR UML EN LOS PROYECTOS. Nathalie López, Jorge Migueis, Emmanuel Picho, 2000
- PIENSA EN JAVA 4ª EDICIÓN. Bruce Eckel, 2008
- CODE COMPLETE. Steve McConnell.
- AJAX. VVAA 2006
- BEGINNING JAVASCRIPT. Paul Wilton. 2000.
- [www.javaranch.com](http://www.javaranch.com)
- [www.c2.com](http://www.c2.com)
- [httpd.apache.org](http://httpd.apache.org)
- [www.javaworld.com](http://www.javaworld.com)
- [www.download.oracle.com/javase/1.4.2/docs/api/](http://www.download.oracle.com/javase/1.4.2/docs/api/)
- [tomcat.apache.org](http://tomcat.apache.org)



## ANEXO I. MANUAL DE USUARIO

### 1. Pantalla principal

la bolsa de la liga Inversores  
temporada 2011/12

**NUEVO Precio**  
1<sup>ra</sup> bolsa 32€  
Sigüientes bolsas 31€/Bolsa  
PRECIO aplicable a cada compra  
Regístrate

**REGÍSTRATE AHORA**  
**COMPRA Y VENDE**  
**GANA DINERO**  
**JUEGA YA**

**COTIZACIONES LIGA BBVA**  
18:09:43

Equipo	Cotización
At. Madrid	1,58
Athletic	1,30
Barcelona	3,90
Betis	0,64
Espanyol	0,78
Getafe	0,78
Granada	0,58
Levante	0,52
Málaga	1,58
Mallorca	0,58
Osasuna	0,54
Racing	0,51
Rayo	0,47
R. Madrid	3,88
R. Sociedad	0,52
Sevilla	1,54
Sporting	0,52
Valencia	1,81
Villarreal	1,72
Zaragoza	0,54

**¿Cómo juego?**  
Instrucciones del juego.  
Resuelve todas tus dudas.

**Compra / venta**  
Desde tu bolsa, puedes comprar y vender acciones de los equipos todas las veces que quieras sin coste adicional.

**Tu dinero seguro**  
El importe de cada bolsa queda ingresado en una cuenta de labolsadelaliga.com en el BBVA.

**Noticias de actualidad**  
Thomson trae grandes recuerdos al Villarreal  
Costinha considera que el Barcelona no es imbatible  
El rey ha muerto, viva el rey

**LIGA BBVA.com**  
Haz clic para ver cómo se hizo el spot de Iniesta y Casillas: la liga que mueve al mundo

Cuando el usuario accede a [www.labolsadelaliga.com/liga-bbva/](http://www.labolsadelaliga.com/liga-bbva/) lo primero que ve es esta página que le da información sobre la cotización de los diferentes equipos, noticias, artículos de opinión...

Desde la parte superior se puede acceder a las diferentes secciones de la parte externa, cada una con información referente a la competición o con consejos sobre el juego.

También en la parte superior se encuentra el formulario de acceso si ya es un usuario registrado o el acceso al formulario de registro si aún no lo es.

## 2. Registro

### 2.1. Formulario de registro

Si el usuario hace clic en el botón “¡Regístrate!” se abrirá la siguiente ventana:

The registration form is titled '¡Regístrate!' and is for 'la bolsa de la liga Inversores'. It contains the following fields and options:

- Nombre: Text input field.
- Apellidos: Text input field.
- Ciudad o población: Text input field.
- Código Postal: Text input field.
- País: Dropdown menu with 'Selecciona un país'.
- Zona horaria: Dropdown menu with 'Selecciona'.
- Fecha de nacimiento: Three dropdown menus for 'Día', 'Mes', and 'Año'.
- Sexo: Radio buttons for 'Hombre' and 'Mujer'.
- Teléfono: Text input field with an 'Opcional' checkbox.
- ¿Cómo nos conociste?: Dropdown menu with 'Anuncio de internet'.
- Email: Text input field.
- Nombre de usuario: Text input field.
- Contraseña: Text input field.
- Repetir contraseña: Text input field.
- Clave de firma: Text input field.
- Repetir clave de firma: Text input field.
- Pregunta de seguridad: Dropdown menu with 'Nombre de mi equipo favorito'.
- Respuesta: Text input field.
- Escribe los caracteres que ves en la imagen siguiente: Image with text 'k6w3d' and a 'Recargar' button.
- Checkboxes: 'Deseo recibir bonos, descuentos, ofertas exclusivas y formar parte en diferentes sorteos' and 'Acepto los [Términos y condiciones](#) y la [Privacidad de datos](#)'.
- Footer: 'Recuerda: Para comenzar a jugar debes comprar 1 bolsa de la Liga BBVA y/o de la Liga adelante.', 'VeriSign Trusted' logo, 'Si tienes un código promocional o de invitación introdúcelo aquí:', and a 'Siguiente' button.

Al rellenar cada campo, un icono indica si el valor introducido es válido o no:

The registration form is filled with sample data, and each field has a green checkmark icon indicating it is valid:

- Nombre: Javier ✓
- Apellidos: Lapieza ✓
- Ciudad o población: Pamplona ✓
- Código Postal: 31004 ✓
- País: Selecciona un país ✓
- Zona horaria: GMT +01 Madrid, París, Berlín ✓
- Fecha de nacimiento: 1 Mayo 1990 ✓
- Sexo: Hombre (selected) ✓
- Teléfono: Opcional ✓
- ¿Cómo nos conociste?: Anuncio de internet ✓
- Email: ✓
- Nombre de usuario: ✓
- Contraseña: ✓
- Repetir contraseña: ✓
- Clave de firma: ✓
- Repetir clave de firma: ✓
- Pregunta de seguridad: Nombre de mi equipo favorito ✓
- Respuesta: ✓
- Escribe los caracteres que ves en la imagen siguiente: Image with text 'ggf7m' and a 'Recargar' button.
- Checkboxes: 'Deseo recibir bonos, descuentos, ofertas exclusivas y formar parte en diferentes sorteos' (unchecked) and 'Acepto los [Términos y condiciones](#) y la [Privacidad de datos](#)' (checked).
- Footer: 'Recuerda: Para comenzar a jugar debes comprar 1 bolsa de la Liga BBVA y/o de la Liga adelante.', 'VeriSign Trusted' logo, 'Si tienes un código promocional o de invitación introdúcelo aquí:', and a 'Siguiente' button.

En este ejemplo todos los datos introducidos son correctos, en el siguiente se indica con colores diferentes los valores correctos y los incorrectos. En el caso de que haya valores incorrectos, se le indicará al usuario mediante un mensaje además de los iconos correspondientes.



Una vez completado con éxito el proceso de registro se le enviará al correo indicado un mensaje para que el usuario confirme que este correo existe haciendo clic sobre un enlace. Este enlace le llevará directamente a la capa de adquisición de bolsas a no ser que el usuario haya introducido un código promocional correcto en la capa de registro, en cuyo caso se puede saltar este paso.

## 2.2. Adquisición de bolsas

Una vez registrado, el usuario debe adquirir al menos una bolsa para poder comenzar a jugar. Una bolsa se compone de 1 acción de cada equipo más una cantidad en efectivo que depende de la división en la que se esté jugando (7,15€ en la liga BBVA y 6,19 en la liga adelante).

Contenido			
	Gastos administrativos para toda la temporada 2 €	20 acciones (una de cada equipo) Valor de las acciones: 22,85€	Efectivo disponible 7,15 €
		+	30,00 €
	Gastos administrativos para toda la temporada 2 €	22 acciones (una de cada equipo) Valor de las acciones: 23,81€	Efectivo disponible 6,19 €
		+	30,00 €

El usuario tiene que completar el siguiente formulario:

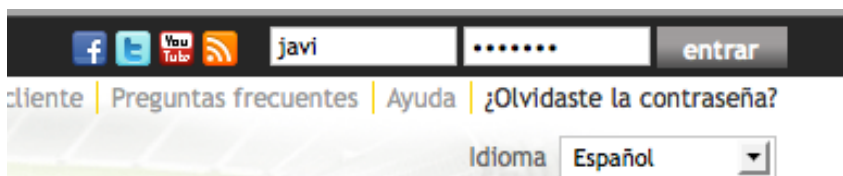
Si no hay errores (números negativos o letras en las casillas), se cargará la pantalla del banco:

El usuario debe completar ahora el resto de datos de su tarjeta de crédito y una vez que el banco los confirme podrá jugar a labolsadelaliga.

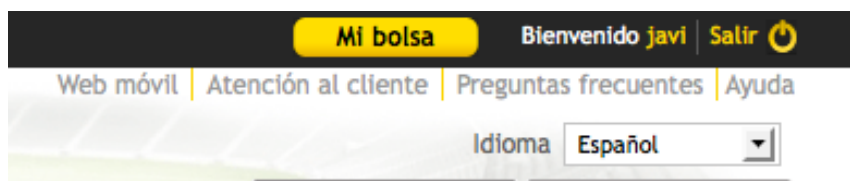
### 3. Acceso

#### 3.1. Formulario de acceso

Para realizar el acceso, los usuarios tendrán que rellenar el formulario de acceso que está en la parte superior derecha de la página. Si los datos son correctos el usuario será redirigido automáticamente a “Mi bolsa”.

Captura de pantalla del formulario de acceso en la parte superior derecha de la página. Incluye iconos de redes sociales (Facebook, Twitter, YouTube, RSS), campos de usuario (javi) y contraseña (puntos), un botón 'entrar', y enlaces a 'cliente', 'Preguntas frecuentes', 'Ayuda' y '¿Olvidaste la contraseña?'. También hay un selector de idioma con 'Español' seleccionado.

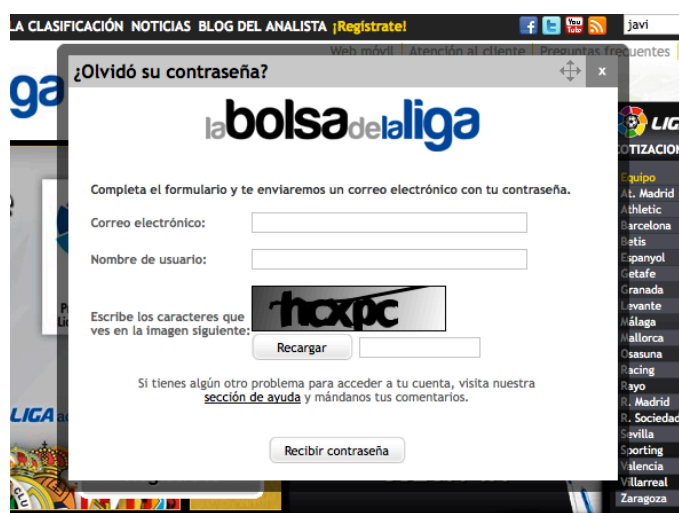
Si el usuario ya tuviese una sesión activa, en lugar de ver el formulario de acceso verá los siguientes botones:

Captura de pantalla de la barra superior de la página cuando el usuario ya tiene una sesión activa. Muestra un botón amarillo 'Mi bolsa', el texto 'Bienvenido javi' y un botón 'Salir' con un icono de apagado. Debajo hay enlaces a 'Web móvil', 'Atención al cliente', 'Preguntas frecuentes' y 'Ayuda', y un selector de idioma con 'Español' seleccionado.

El usuario podrá acceder a su bolsa haciendo clic en el botón correspondiente o sobre su nombre de usuario.

#### 3.2. Recuperar contraseña

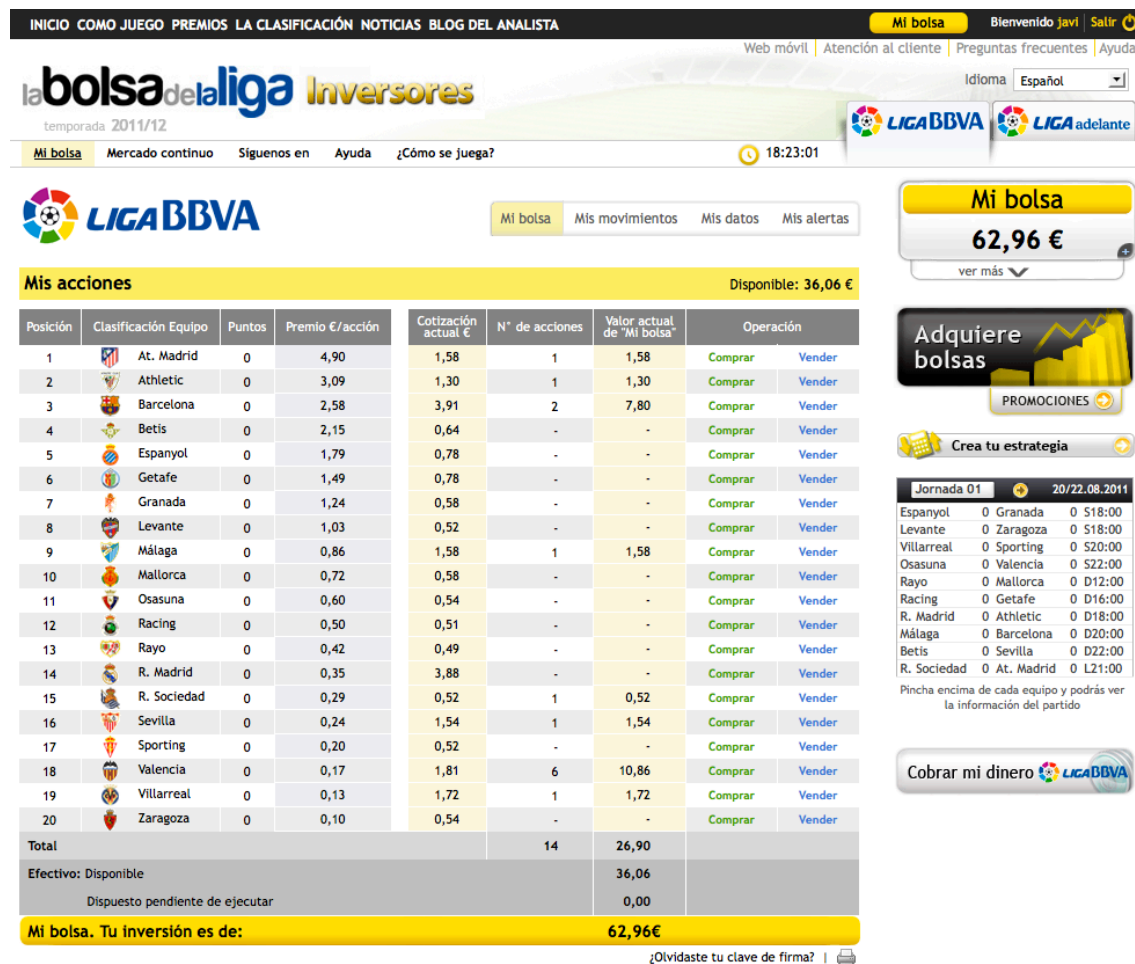
Si el usuario no recuerda su contraseña, puede hacer clic en “¿Olvidó su contraseña?” y le aparecerá el siguiente formulario, que podrá completar para recibir una nueva contraseña en su correo electrónico.

Captura de pantalla del formulario de recuperación de contraseña que aparece al hacer clic en '¿Olvidó su contraseña?'. El formulario pide completar los datos para enviar un correo electrónico con la nueva contraseña. Incluye campos para 'Correo electrónico:', 'Nombre de usuario:', y un campo para escribir los caracteres que aparecen en una imagen (hoxpc). Hay botones 'Recargar' y 'Recibir contraseña'. En la parte inferior, hay un enlace a la 'sección de ayuda' y una invitación a mandar comentarios si hay algún problema.

Sólo se enviará una nueva contraseña si los datos introducidos coinciden con los de nuestra base de datos.

## 4. Mi bolsa

Esta es la sección principal del juego ya que es donde se desarrolla el mismo. Desde aquí el usuario tiene acceso a todas las secciones relacionadas con el juego y puede operar con sus acciones.



**la bolsa de la liga Inversores**  
temporada 2011/12

**Mi bolsa** Mercado continuo Siguenos en Ayuda ¿Cómo se juega?

**LIGABBVA**

**Mi bolsa** Mis movimientos Mis datos Mis alertas

**Mi bolsa**  
62,96 €  
ver más

**Adquiere bolsas**  
PROMOCIONES

**Crea tu estrategia**

**Jornada 01** 20/22.08.2011

Equipo	Puntos	Premio €/acción	Cotización actual €	Nº de acciones	Valor actual de "Mi bolsa"	Operación
At. Madrid	0	4,90	1,58	1	1,58	Comprar Vender
Athletic	0	3,09	1,30	1	1,30	Comprar Vender
Barcelona	0	2,58	3,91	2	7,80	Comprar Vender
Betis	0	2,15	0,64	-	-	Comprar Vender
Espanyol	0	1,79	0,78	-	-	Comprar Vender
Getafe	0	1,49	0,78	-	-	Comprar Vender
Granada	0	1,24	0,58	-	-	Comprar Vender
Levante	0	1,03	0,52	-	-	Comprar Vender
Málaga	0	0,86	1,58	1	1,58	Comprar Vender
Mallorca	0	0,72	0,58	-	-	Comprar Vender
Osasuna	0	0,60	0,54	-	-	Comprar Vender
Racing	0	0,50	0,51	-	-	Comprar Vender
Rayo	0	0,42	0,49	-	-	Comprar Vender
R. Madrid	0	0,35	3,88	-	-	Comprar Vender
R. Sociedad	0	0,29	0,52	1	0,52	Comprar Vender
Sevilla	0	0,24	1,54	1	1,54	Comprar Vender
Sporting	0	0,20	0,52	-	-	Comprar Vender
Valencia	0	0,17	1,81	6	10,86	Comprar Vender
Villarreal	0	0,13	1,72	1	1,72	Comprar Vender
Zaragoza	0	0,10	0,54	-	-	Comprar Vender
<b>Total</b>				<b>14</b>	<b>26,90</b>	
<b>Efectivo: Disponible</b>					<b>36,06</b>	
<b>Dispuesto pendiente de ejecutar</b>					<b>0,00</b>	

**Mi bolsa. Tu inversión es de:** 62,96€

¿Olvidaste tu clave de firma?

### 4.1 El menú

El menú superior tiene acceso a las diferentes secciones de “Mi bolsa”: “Mis movimientos”, donde el usuario puede consultar todas sus compraventas, tanto realizadas como pendientes; “Mis datos”, para configurar su cuenta o cambiar sus datos personales y “Mis Alertas”, donde los usuarios pueden configurar diferentes alertas para recibirlas en su correo electrónico.



**la bolsa de la liga Inversores**  
temporada 2011/12

**Mi bolsa** Mercado continuo Siguenos en Ayuda ¿Cómo se juega?

**LIGABBVA**

**Mi bolsa** Mis movimientos Mis datos Mis alertas

**Mi bolsa**  
62,97 €

Al lado del menú se puede ver el valor actual de nuestra bolsa, este valor se calcula con relación a la cotización de cada equipo, por lo que puede variar a lo largo del tiempo aunque el usuario no realice operaciones. Se puede desplegar para ver un desglose del mismo.

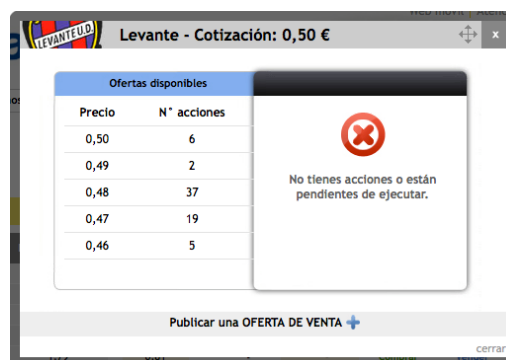
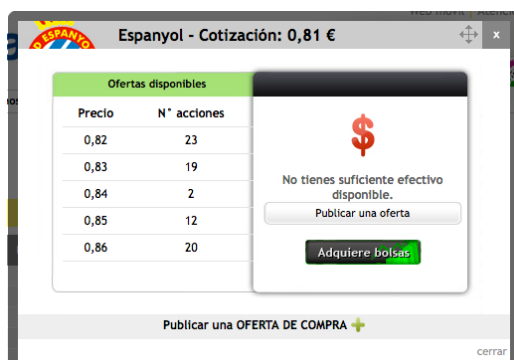


## 4.2 Comprar y vender acciones

Junto a cada equipo están los botones de comprar y vender, al hacer clic sobre uno de ellos se abre la capa de operaciones.



La capa de operaciones solo permite ejecutar las ofertas más beneficiosas para el usuario, cuando decidas comprar o vender siempre lo harás al mejor precio disponible. En la parte inferior hay una barra que va disminuyendo de tamaño. Es el tiempo disponible para realizar la operación. Si el usuario no pudiese realizar la operación solicitada, se le mostrará un mensaje.



Si las ofertas existentes no le son satisfactorias el usuario puede publicar una oferta propia haciendo clic en “publicar una oferta”.

Precio	N° acciones
3,90	13
3,89	15
3,88	7
3,87	13
3,86	1

Volver a OFERTAS EXISTENTES

Precio de venta:  N° de acciones a vender:  Importe operación:  Vigencia de la oferta:

Clave de firma:  [¿Olvidaste tu clave?](#)

La oferta tiene que cumplir una serie de reglas para ser válida:

- Tiene que estar entre los valores máximos y mínimos de premio.
- Tiene que ser más beneficiosa para el usuario que las ya existentes.
- El usuario tiene que tener suficientes acciones o efectivo para satisfacer la oferta en el caso de que ésta fuese ejecutada.

Una vez publicada la oferta las acciones o el dinero comprometidos no estarán disponibles hasta que la oferta sea retirada.

#### 4.3. Pantallas de equipo

Haciendo clic sobre un equipo se puede ver toda la información relativa al mismo, cotización, resultados, calendario... junto con una gráfica que muestra la evolución de su cotización a lo largo de la temporada.

**Cotización de equipo**

At. Osasuna

Hora: **20:05:26**
Cotización actual: **0.54 €**

Temporada 2011/12

Jornada 1

Comprar

Vender

**PARTIDOS - Temporada 2011/12**

■ Victoria
■ Empate
■ Derrota

PRIMERA VUELTA				SEGUNDA VUELTA			
Jornada	Equipo 1	Equipo 2	Resultado	Jornada	Equipo 1	Equipo 2	Resultado
01	Osasuna	Valencia	- -	20	Valencia	Osasuna	- -
02	At. Madrid	Osasuna	- -	21	Osasuna	At. Madrid	- -
03	Osasuna	Sporting	- -	22	Sporting	Osasuna	- -
04	Barcelona	Osasuna	- -	23	Osasuna	Barcelona	- -
05	Osasuna	Sevilla	- -	24	Sevilla	Osasuna	- -
06	Granada	Osasuna	- -	25	Osasuna	Granada	- -
07	Osasuna	Mallorca	- -	26	Mallorca	Osasuna	- -

#### 4.4. Recuperar clave de firma

Para realizar cualquier operación en la capa de operaciones el usuario tendrá que validarla con su clave de firma. Si ésta se le olvida, puede recurrir a un formulario similar al de “¿Olvidó su contraseña?” para recuperarla. Siempre que se le solicite la clave de firma el usuario tendrá la opción de abrir este formulario.

Completa el formulario y te enviaremos un correo electrónico con tu nueva clave de firma.

Correo electrónico:

Nombre de usuario:

Contraseña de usuario:

Escribe los caracteres que ves en la imagen siguiente:

Si tienes algún otro problema para recuperar tu clave de firma, visita nuestra [sección de ayuda](#) y mándanos tus comentarios.

## 5. Mis movimientos

En esta sección el usuario puede consultar todos sus movimientos realizados o pendientes, ordenarlos por equipo, por fechas... haciendo clic en la ventana correspondiente.

Ofertas pendientes:



[Mi bolsa](#)
[Mis movimientos](#)
[Mis datos](#)
[Mis alertas](#)

Acciones pendientes de ejecutar
Acciones compradas
Acciones vendidas

Todos

Disponibles: **36,06€**

**En curso**

Operación	Equipo	Cotización actual €	Precio €	Nº de acciones	Importe operación €	Vigencia de la oferta
Venta	Barcelona	3,91	4,15	1	4,15	28/08/2011 11:30 <span style="float: right;">retirar</span>

Compras realizadas:

Acciones pendientes de ejecutar
Acciones compradas
Acciones vendidas

Todos

Buscar desde:

Hasta:

Buscar

**Acciones compradas**

Operación	Equipo	Premio €/acción	Precio €	Nº de acciones	Importe operación €	Fecha operación
Compra	Valencia	0,17	1,84	4	7,36	21/08/2011 18:44:51
Compra	Valencia	0,17	1,80	1	1,80	19/08/2011 16:23:50
Compra	Barcelona	2,58	3,76	1	3,76	19/08/2011 16:22:57
Compra	R. Madrid	0,35	3,67	5	18,35	19/08/2011 16:21:29
Compra	R. Madrid	0,35	3,66	2	7,32	19/08/2011 16:21:20
Compra	Granada	1,24	0,37	18	6,66	19/08/2011 16:20:08

1



Ventas realizadas:

Acciones pendientes de ejecutar

Acciones compradas

Acciones vendidas

Todos

Buscar desde:

Hasta:

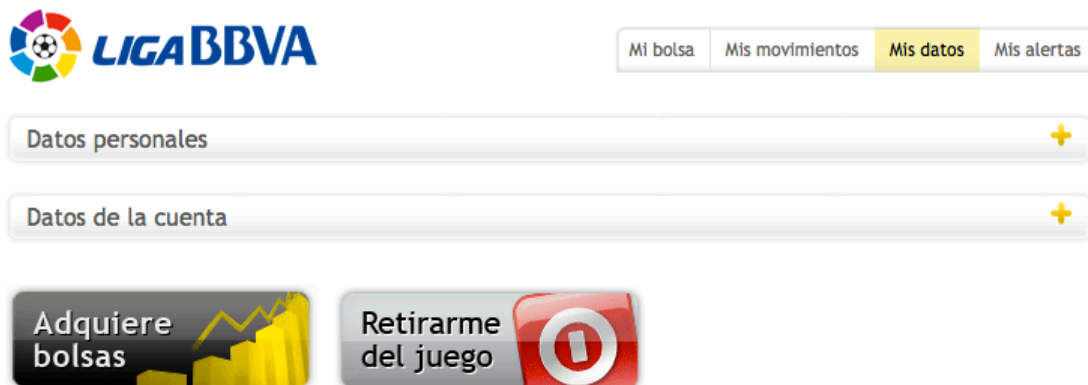
Buscar

Acciones vendidas

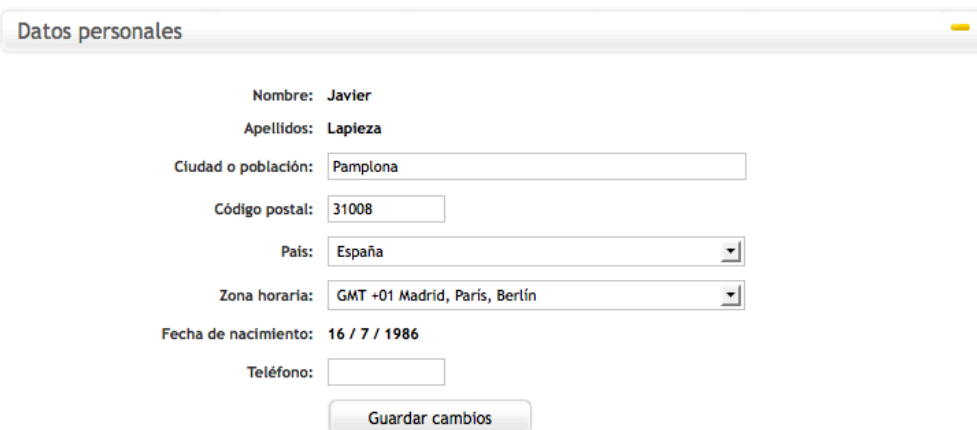
Operación	Equipo	Premio €/acción	Precio €	Nº de acciones	Importe operación €	Fecha operación
Venta	Espanyol	1,79	0,73	1	0,73	23/08/2011 08:59:58
Venta	Getafe	1,49	0,77	1	0,77	23/08/2011 08:59:43
Venta	Zaragoza	0,10	0,50	1	0,50	22/08/2011 19:06:09
Venta	Granada	1,24	0,44	3	1,32	22/08/2011 09:01:25
Venta	Levante	1,03	0,45	1	0,45	22/08/2011 08:26:18
Venta	Granada	1,24	0,42	16	6,72	21/08/2011 11:37:14
Venta	Mallorca	0,72	0,56	1	0,56	21/08/2011 11:36:38
Venta	R. Madrid	0,35	3,86	5	19,30	20/08/2011 00:42:03
Venta	R. Madrid	0,35	3,85	0	0,00	20/08/2011 00:40:16
Venta	R. Madrid	0,35	3,85	0	0,00	20/08/2011 00:40:15
Venta	R. Madrid	0,35	3,85	0	0,00	20/08/2011 00:40:14
Venta	R. Madrid	0,35	3,85	0	0,00	20/08/2011 00:40:11
Venta	R. Madrid	0,35	3,85	3	11,55	20/08/2011 00:40:00
Venta	Rayo	0,42	0,34	1	0,34	19/08/2011 16:24:52
Venta	Sporting	0,20	0,49	1	0,49	19/08/2011 16:24:27
Venta	Racing	0,50	0,35	1	0,35	19/08/2011 16:24:08
Venta	Betis	2,15	0,58	1	0,58	19/08/2011 16:23:16
Venta	Osasuna	0,60	0,50	1	0,50	19/08/2011 12:28:48

## 6. Mis datos

Desde esta sección, el usuario puede consultar y cambiar sus datos personales o de su cuenta, puede abrir cada sección haciendo clic sobre el título correspondiente.



### Datos personales:



Datos de la cuenta:

Datos de la cuenta

### Elige tu juego de inicio

la **bolsa** de la **liga** Inversores

Guardar cambios

### Usuario

E-mail: javier@labolsadelaliga.com  
Nombre de usuario: javi

### Contraseña

Contraseña actual:

Contraseña nueva:  ?

Nivel de seguridad ■

Repetir contraseña nueva:

Guardar cambios

### Clave de firma

Clave de firma actual:

Clave de firma nueva:  ?

Nivel de seguridad ■

Repetir clave de firma nueva:

Guardar cambios

El inicio preferido indica que bolsa quiere ver el usuario por defecto cuando acceda al sistema, basta con indicarlo en el formulario y hacer clic en “Guardar cambios”. Para realizar cambios en contraseña o clave de firma siempre habrá que introducir la clave anterior.

Además en esta sección está el botón de retirarse del juego, este proceso es análogo al de “Cobrar mi dinero” que se explicará más adelante, con la diferencia de que el usuario es además retirado del sistema.

## 7. Mis Alertas

En esta sección el usuario puede ver y configurar las alertas de correo que quiere que le sean enviadas. Por defecto se envían alertas siempre que nos compren o nos vendan acciones.



Mi bolsa   Mis movimientos   Mis datos   **Mis alertas**

**Crear una alerta**

En esta página puedes crear las alertas que te ayuden a jugar. Tú puedes generar cuantas quieras, modificarlas o eliminarlas. Siempre te comunicaremos cuando se produzca una compra-venta de tus acciones.

### Listado de mis alertas

#### Compra/venta de mis acciones

		Fecha de creación	
Todos	Vendan acciones	25/08/2010	Retirar
Todos	Compren acciones	25/08/2010	Retirar

**Crear una alerta**

#### Caducidad de mis ofertas

Fecha de creación

**Crear una alerta**

#### Acciones que cotizan más/menos de...

Fecha de creación

**Crear una alerta**

#### Sube o baja la cotización

Fecha de creación

**Crear una alerta**

#### Avisarme vía email cuando haya nuevas promociones

Fecha de creación

**Crear una alerta**

Para retirar una oferta basta con hacer clic en “Retirar” e introducir la clave de firma en el formulario correspondiente.

Para crear alertas, hay que hacer clic en el tipo de alertas que se quiere crear y se completa el formulario con los datos necesarios.

**Acciones coticen más/menos de...**

Selecciona un equipo:

At. Madrid	<input type="checkbox"/>
Athletic	<input type="checkbox"/>
Barcelona	<input checked="" type="checkbox"/>
Betis	<input type="checkbox"/>
Espanyol	<input type="checkbox"/>
Getafe	<input type="checkbox"/>
Granada	<input type="checkbox"/>
Levante	<input type="checkbox"/>
Málaga	<input type="checkbox"/>

Acciones coticen más de

Acciones coticen menos de

**Crear alerta**

Todas las alertas, excepto las de compras y ventas se eliminan tras el primer envío para evitar un envío excesivo de correos al usuario.

## 8. Crea tu estrategia

Esta herramienta permite al usuario calcular como quedaría la clasificación en función de sus pronósticos, es accesible en un botón a la derecha en la sección “Mi bolsa”.



Al hacer clic, aparece la capa de “Crea tu estrategia”, con las dos siguientes jornadas por disputar. El usuario debe rellenar cada jornada por completo para poder realizar el cálculo. Puede realizar el cálculo jornada a jornada o ir rellenando todas las jornadas que quiera para realizar un pronóstico más amplio. Para poder pasar una jornada siempre hay que rellenar la anterior.

The screenshot shows a web interface titled 'Crea tu estrategia de juego en la Liga BBVA'. It features a table of team rankings on the left, a central area for selecting match outcomes for 'Jornada 02' and 'Jornada 03', and buttons for 'Borrar datos' and 'Actualiza tabla'. The interface includes the Liga BBVA logo and navigation links at the top.

Posición	Clasificación	Equipo	Puntos	R.J.	Premio €/acción
1		At. Madrid	0	0	4,90
2		Athletic	0	0	3,09
3		Barcelona	0	0	2,58
4		Betis	0	0	2,15
5		Espanyol	0	0	1,79
6		Getafe	0	0	1,49
7		Granada	0	0	1,24
8		Levante	0	0	1,03
9		Málaga	0	0	0,86
10		Mallorca	0	0	0,72
11		Osasuna	0	0	0,60
12		Racing	0	0	0,50
13		Rayo	0	0	0,42
14		R. Madrid	0	0	0,35
15		R. Sociedad	0	0	0,29
16		Sevilla	0	0	0,24
17		Sporting	0	0	0,20
18		Valencia	0	0	0,17
19		Villarreal	0	0	0,13
20		Zaragoza	0	0	0,10

Temporada: 2011/12    Jornada: 2    Clasificación: VIRTUAL    Jornada: 2

Aquí tienes una herramienta que te ayudará a crear tu propia estrategia de juego. Puedes simular los resultados de una jornada de liga o todas las que quieras. Al "Calcular hasta aquí" puedes utilizar las jornadas que tu quieras y realizar dos cálculos a la vez. Cuando hagas click sobre "Actualizar tabla", el sistema te dará "tu clasificación virtual", la que tú has creado con tu simulación.

**Jornada 02** 20/22.08.2011

	1	X	2
Sporting	R. Sociedad	<input type="radio"/>	<input type="radio"/>
Valencia	Racing	<input type="radio"/>	<input type="radio"/>
Granada	Betis	<input type="radio"/>	<input type="radio"/>
At. Madrid	Osasuna	<input type="radio"/>	<input type="radio"/>
Athletic	Rayo	<input type="radio"/>	<input type="radio"/>
Mallorca	Espanyol	<input type="radio"/>	<input type="radio"/>
Getafe	Levante	<input type="radio"/>	<input type="radio"/>
Zaragoza	R. Madrid	<input type="radio"/>	<input type="radio"/>
Sevilla	Málaga	<input type="radio"/>	<input type="radio"/>
Barcelona	Villarreal	<input type="radio"/>	<input type="radio"/>

Calcular hasta aquí

**Jornada 03** 20/22.08.2011

	1	X	2
Valencia	At. Madrid	<input type="radio"/>	<input type="radio"/>
Osasuna	Sporting	<input type="radio"/>	<input type="radio"/>
R. Sociedad	Barcelona	<input type="radio"/>	<input type="radio"/>
Villarreal	Sevilla	<input type="radio"/>	<input type="radio"/>
Málaga	Granada	<input type="radio"/>	<input type="radio"/>
Betis	Mallorca	<input type="radio"/>	<input type="radio"/>
Espanyol	Athletic	<input type="radio"/>	<input type="radio"/>
Rayo	Zaragoza	<input type="radio"/>	<input type="radio"/>
R. Madrid	Getafe	<input type="radio"/>	<input type="radio"/>
Racing	Levante	<input type="radio"/>	<input type="radio"/>

Calcular hasta aquí

Borrar datos    Actualiza tabla

Si falta algún partido por completar el sistema avisará mediante un mensaje y marcará en rojo los partidos sin pronóstico.

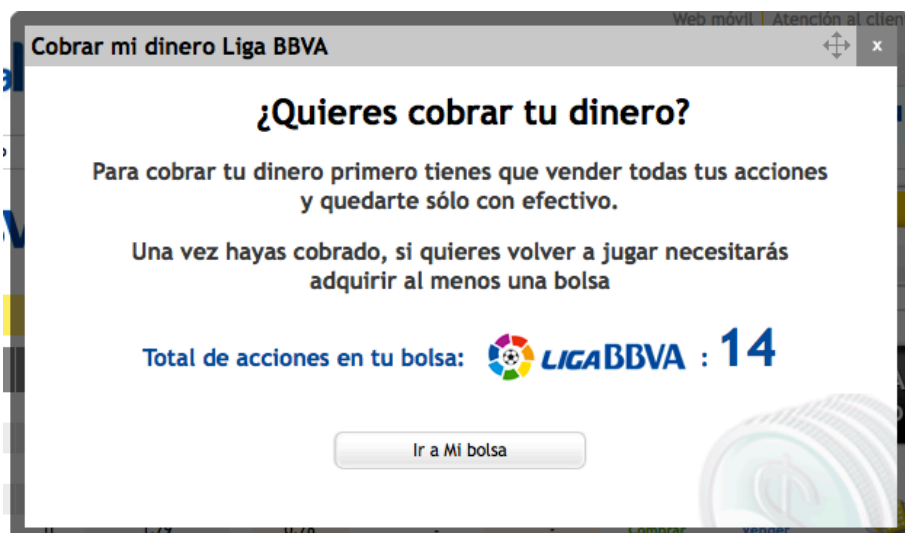
This image is a close-up of the match selection interface for 'Jornada 02'. It shows a list of matches with radio buttons for selecting the outcome (1, X, or 2). The match 'Getafe vs Levante' is highlighted in red, indicating it is a match without a forecast.

	1	X	2
Sporting	R. Sociedad	<input type="radio"/>	<input type="radio"/>
Valencia	Racing	<input type="radio"/>	<input type="radio"/>
Granada	Betis	<input type="radio"/>	<input type="radio"/>
At. Madrid	Osasuna	<input type="radio"/>	<input type="radio"/>
Athletic	Rayo	<input type="radio"/>	<input type="radio"/>
Mallorca	Espanyol	<input type="radio"/>	<input type="radio"/>
Getafe	Levante	<input type="radio"/>	<input type="radio"/>
Zaragoza	R. Madrid	<input type="radio"/>	<input type="radio"/>
Sevilla	Málaga	<input type="radio"/>	<input type="radio"/>
Barcelona	Villarreal	<input type="radio"/>	<input type="radio"/>

Calcular hasta aquí

## 9. Cobrar mi dinero

Para poder retirarse del juego un usuario tiene que vender previamente todas sus acciones, si aún le quedan acciones por vender y hace clic en “Cobrar mi dinero” recibirá información sobre las mismas.




Web móvil | Atención al cliente

**Cobrar mi dinero Liga BBVA**

**¿Quieres cobrar tu dinero?**

Para cobrar tu dinero primero tienes que vender todas tus acciones y quedarte sólo con efectivo.

Una vez hayas cobrado, si quieres volver a jugar necesitarás adquirir al menos una bolsa

Total de acciones en tu bolsa:  **LIGABBVA : 14**

[Ir a Mi bolsa](#)

Si, por el contrario, ya ha vendido todas sus acciones, tendrá acceso al formulario de solicitudes de cobro. En este formulario debe introducir los datos de la cuenta donde desea recibir el ingreso. Si el número de cuenta no tiene el formato correcto el usuario recibirá un mensaje para que lo corrija. Una vez la petición sea realizada, el usuario recibirá el dinero en un plazo de 10 días hábiles.



Web móvil | Atención al cliente

**Cobrar mi dinero Liga BBVA**

**¿Quieres cobrar tu dinero?**

Has vendido todas tus acciones de la Liga BBVA  
Completa tus datos de la cuenta bancaria y podrás cobrar tu dinero

Nombre:

Apellidos:

Reintegro: - €

**España** Resto de mundo

Introduce tu número de cuenta CCC con los 20 dígitos:

Para terminar, introduce tu Clave de firma:

[Retirar mi dinero](#)

\* Si eres jugador de la Temporada 2010/11 tienes de plazo hasta el 25 de agosto para poder retirar tu dinero, deberás hacerlo a través del servicio de atención al cliente.

## 10. Mercado Continuo

El mercado continuo proporciona información del estado del mercado a los usuarios. Además de la cotización muestra el número de acciones en oferta de cada equipo así como el número de operaciones realizadas sobre el mismo.



Posición	Clasificación equipo	Puntos	Premio €/acción	Cotización actual €	Operaciones	
					Realizadas	Pendientes
1	At. Madrid	0	4,90	1,58	361	455
2	Athletic	0	3,09	1,30	518	436
3	Barcelona	0	2,58	3,91	351	335
4	Betis	0	2,15	0,64	613	921
5	Espanyol	0	1,79	0,78	497	1.435
6	Getafe	0	1,49	0,78	904	495
7	Granada	0	1,24	0,58	3.393	465
8	Levante	0	1,03	0,52	620	802
9	Málaga	0	0,86	1,58	602	856
10	Mallorca	0	0,72	0,58	643	457
11	Osasuna	0	0,60	0,54	407	486
12	Racing	0	0,50	0,51	1.074	1.096
13	Rayo	0	0,42	0,49	1.573	352
14	R. Madrid	0	0,35	3,88	411	350
15	R. Sociedad	0	0,29	0,52	644	403
16	Sevilla	0	0,24	1,55	332	290
17	Sporting	0	0,20	0,52	735	742
18	Valencia	0	0,17	1,81	296	344
19	Villarreal	0	0,13	1,72	470	386
20	Zaragoza	0	0,10	0,54	728	711

## 11. Ayuda

Los usuarios tienen una amplia sección de ayuda donde consultar sus dudas. O enviar consultas a nuestro correo electrónico.

### 11.1 Preguntas frecuentes

Preguntas más frecuentes sobre el juego, incluye un buscador y un formulario para enviar tu duda si no encuentras respuesta.

Web móvil | Atención al cliente | F

la **bolsa** de la **liga** Inversores

temporada 2011/12

## Preguntas frecuentes

Buscar

- 1. Como juego (17)
- 2. Mi cuenta (3)
- 3. Privacidad del usuario (3)
- 4. Información bancaria (5)

No encuentro lo que buscaba, quiero enviaros una pregunta:

Correo electrónico:

Escribe los caracteres que ves en la imagen:

Recargar imagen

Enviar

### • Como juego

¿Cómo puedo obtener más efectivo disponible?

Lógicamente si vendes tus acciones obtendrás más efectivo para poder comprar. Si no quieres vender tus acciones, puedes obtener más efectivo adquiriendo más bolsas.

¿Se puede jugar en peña o grupo?

¿Cómo se obtiene la cotización?

¿Cuál es el importe máximo que puedo vender?

¿Cuál es el la cantidad máxima de bolsas que puedo comprar?

¿Cuándo puedo efectuar acciones de compra-venta?

¿Cómo sé que una compra/venta se ha hecho correctamente?

¿Cómo sé de cuánto dinero dispongo?

¿Cómo sé cuánto estoy ganando?

¿Cuánto puedo perder?

¿Cuánto puedo ganar?

¿Cómo y cuándo recupero lo ganado?

¿Cuánto puedo jugar?

¿Cómo se juega?

¿Cuándo termina el juego?

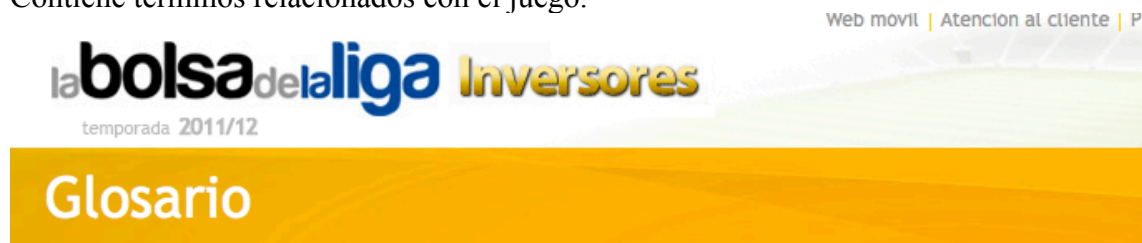
¿Cómo empiezo a jugar?

¿Qué es una bolsa?



## 11.2 Glosario

Contiene términos relacionados con el juego.



 [Buscador](#)

 [Glosario](#)

 [Preguntas frecuentes](#)

 [Como juego](#)

### Glosario

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

#### Bajar

Decimos que un equipo baja la cotización cuando los jugadores de la bolsadelaliga están ofertando acciones a la baja, es decir, descendiendo sobre el precio de referencia que tenía ese equipo.

#### Bloqueo

Espacio de tiempo del que dispones para comprar o vender, durante el cual las acciones que hayas elegido, sólo están disponibles para ti. El tiempo de bloqueo es de 25 segundos, tras lo cual el sistema "liberará" las acciones.

#### Bolsa

Una bolsa se compone de una acción por cada uno de los equipos de la liga en la que juegues, además de dinero en efectivo para comprar más acciones.

#### Borrar

Acción de anular una operación.

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

### 11.3 Atención al cliente

Si el usuario tiene alguna duda que no logra resolver mediante la información ofrecida en la página, puede enviarnos sus consultas a través de esta sección.



## Atención al cliente

Si necesitas ayuda, por favor, contacta con nosotros utilizando uno de los métodos que se listan a continuación:

### Formulario de contacto



Si tienes alguna pregunta, comentario o sugerencia, escribe tu mensaje y tu e-mail y después haz click en Enviar.

Recibirás la respuesta en tu dirección e-mail lo antes posible.

E-mail:

Mensaje:



Recargar imagen

☐ Acepto los [Términos y condiciones](#) y la [Privacidad de datos](#).

Enviar

# PROYECTO DE FIN DE CARRERA

## DESARROLLO DE UNA APLICACIÓN DE COMPRAVENTA DE ACCIONES: LABOLSADELALIGA

Pamplona, 9 de Septiembre de 2011

Alumno: Javier Lapieza Fernández

Tutor: Eduardo Alfaro Larragueta

- Introducción
- Objetivos
- Implementación
- Implantación
- Pruebas
- Conclusiones
- Líneas futuras
- Presencia en medios
- Demostración

# Introducción

- Juegos on-line
  - Apuestas deportivas, juegos relacionados con el deporte...
- Banca on-line
  - Transferencias, compraventa de acciones...
- Este proyecto une estos dos conceptos: las apuestas deportivas y el mercado de valores, la bolsa

# Objetivos

“La implementación de una aplicación de compraventa de acciones de los equipos de fútbol de la liga de fútbol profesional cuyo núcleo pueda exportarse a otras ligas o incluso deportes.”

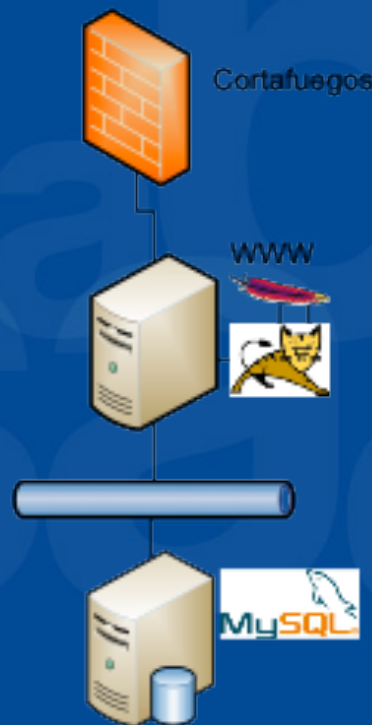
# Objetivos

- Sistema robusto, escalable y eficiente
- Procesar pagos y trabajar con dinero real
- Adaptabilidad
  - Exportable a otras ligas o deportes
- Compatibilidad
  - Llegar al máximo número de usuarios

- Metodologías ágiles
- Java, JavaScript, velocity, html, xml...
- Apache + Tomcat + MySQL
- Software auxiliar para acceso y manipulación de datos
- Sistema de transacciones



# Implantación



## Servidor Apache

Procesa las peticiones de las vistas y componentes AJAX

Redirige las peticiones a los servlets

## Servidor Tomcat:

Aloja los diferentes servlets

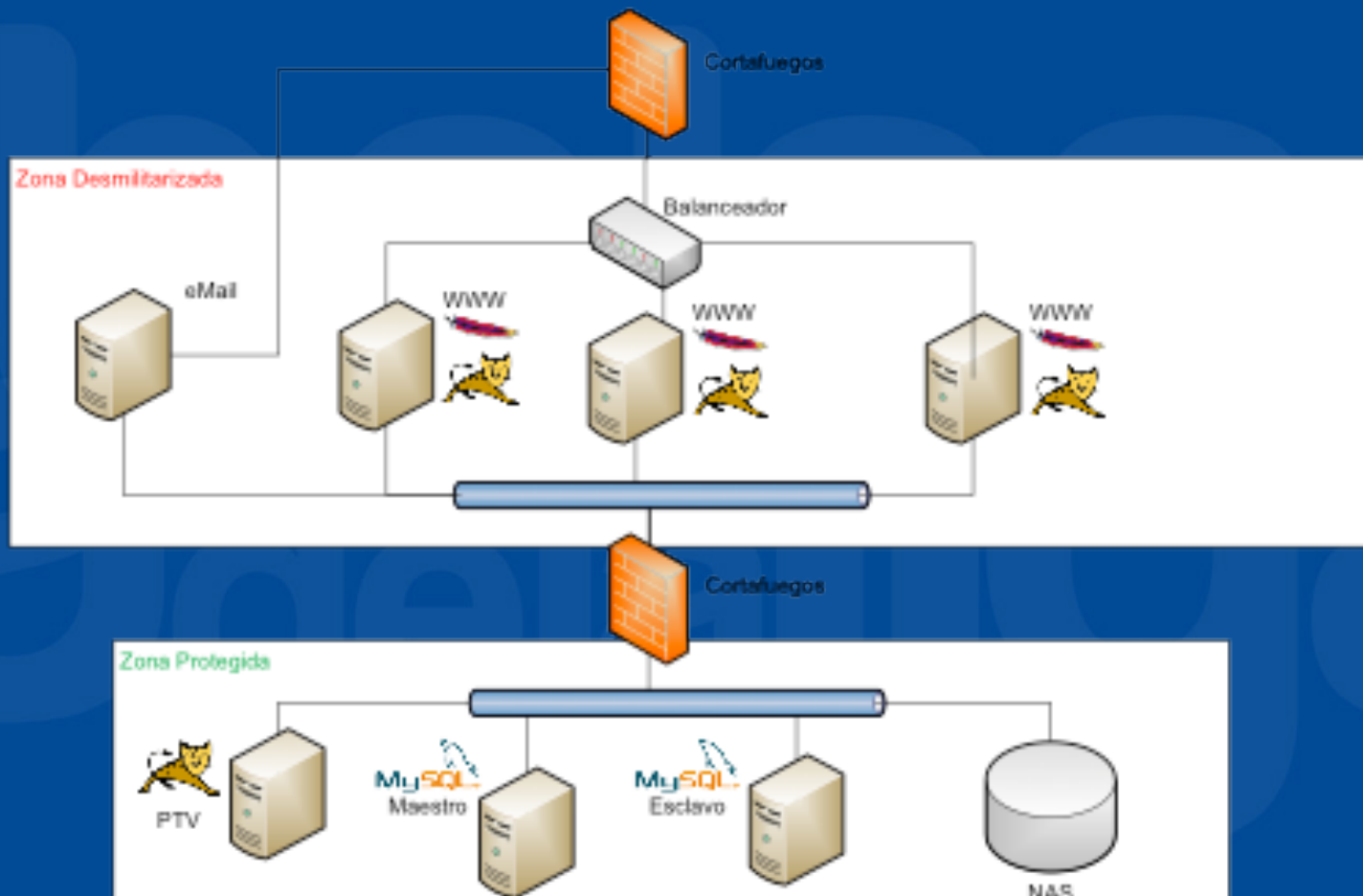
Genera las vistas “velocity”

Gestiona de los datos de sesión

## Servidor MySQL:

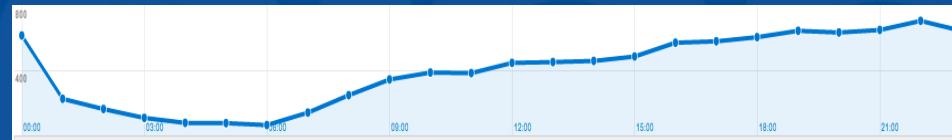
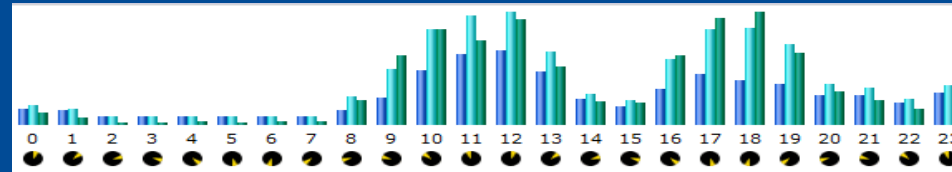
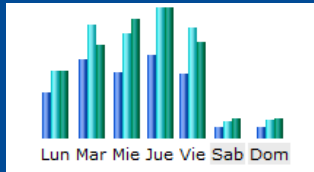
Capa de persistencia de los datos

# Implantación



- Pruebas de carga simuladas
- Pruebas con grupo de usuarios
- Auditoría de seguridad:
  - S21sec
- Auditoría de calidad:
  - Medyction

# Pruebas



Usuarios Registrados	Usuarios Concurrentes Pico	Ancho de banda máximo
100.000	2547	35 Mbits
300.000	16279	90 Mbits
1.000.000	24465	300 Mbits

Usuarios Registrados	Usuarios Concurrentes pico	Ancho de banda máximo
100.000	864	12 MB/s
300.000	2485	36 MB/s
1.000.000	8210	120 MB/s

$$C = n L / T ;$$

$$C_{\max} \approx C + 3 \sqrt{C}, \text{ con una probabilidad de } 99.87\%$$

- El proyecto cumple todos los objetivos
- Ciclo de vida completo del sistema
- Integración de diferentes tecnologías
- Generación de código a partir de metadatos
- Trabajo en equipo
- El 25 de Agosto de 2010 el sistema fue publicado online
  - [www.labolsadelaliga.com](http://www.labolsadelaliga.com)

- Más de 50.000 usuarios registrados
- Usuarios de mas de 30 países
- Cerca de un millón de páginas vistas
- Más de medio millón de correos enviados
- Presencia en diferentes medios:
  - Diario de Navarra
  - Marca
  - Mundo deportivo

- Modularización del sistema
- Implementación de un juego “virtual”
- Implementación del resto de grandes ligas de fútbol
- Implementación de ligas de otros deportes NBA, ACN, NFL...



# la**bolsa**de**la**liga Presencia en medios





[www.labolsadelaliga.com](http://www.labolsadelaliga.com)